版权注意事项：
1、书籍版权归作者和出版社所有
2、本PDF仅限用于个人获取知识，进行私底下的知识交流
3、PDF获得者不得在互联网上以任何目的进行传播
4、如觉得书籍内容很赞，请购买正版实体书，支持作者
5、请于下载PDF后24小时内删除本PDF。

Alibaba Group 阿里巴巴集团 | 技术丛书

Broadview www.broadview.com.cn

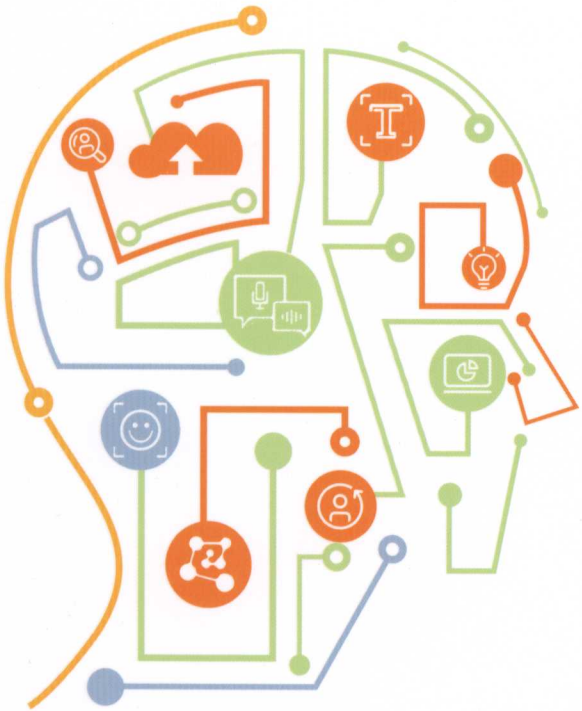
大数据丛书 | “十三五”国家重点图书出版规划项目

永远在线 | 实时响应 | 轻松上手 | 畅学无忧

机器学习在线

解析阿里云机器学习平台

杨旭 著



中国工信出版集团

電子工業出版社 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY http://www.phei.com.cn

Alibaba Group 阿里巴巴集团 | 技术丛书

大数据丛书 | “十三五”国家重点图书出版规划项目

机器学习在线

解析阿里云机器学习平台

杨旭 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

近几年，机器学习平台获得了飞速发展，积累了大量高效的机器学习算法组件，基于这些组件可以快速实现业务流程、解决具体问题。阿里云机器学习平台的丰富算法功能可以在线使用，不需要购买硬件，不需要安装配置各种环境；数据和计算资源一直处在“在线”状态，不必担心数据太大或计算资源不足的问题。机器学习平台降低了我们使用机器学习知识的门槛，将各个算法作为组件，即使不了解背后的理论知识，仍可以仿照书中实例，将组件连接起来解决一些实际问题。

本书适合机器学习算法的初学者及中级用户快速入门，在机器学习实践中学习。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

机器学习在线：解析阿里云机器学习平台 / 杨旭著. —北京：电子工业出版社，2017.8
（阿里巴巴集团技术丛书）
ISBN 978-7-121-31869-6

Ⅰ. ①机… Ⅱ. ①杨… Ⅲ. ①电子商务－计算机网络 Ⅳ. ①TP393

中国版本图书馆 CIP 数据核字(2017)第 131004 号

策划编辑：刘　皎
责任编辑：郑柳洁
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：16.5　字数：263 千字
版　　次：2017 年 8 月第 1 版
印　　次：2017 年 8 月第 1 次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前言

飞速发展的互联网、物联网每时每刻都在产生大量的数据，数据的价值也因此被提升到前所未有的高度：越来越多的人投身数据分析的领域，希望通过机器学习及深度学习，从数据中获取更大的价值。另一方面，云计算的蓬勃发展极大地扩展了数据的存储能力，它使计算可以同时使用成百上千台机器，快速解决问题，而在计算完成后，又能及时释放掉资源，控制成本。

在这样的大背景下，机器学习算法平台也获得了飞速发展，积累了大量高效的机器学习算法组件，基于这些组件我们可以快速实现业务流程，解决具体问题。在为本书定书名时，受到王坚博士《在线》一书的影响，觉得用“在线”一词来说明目前机器学习平台的状态非常恰当：丰富的算法功能可以在线使用、不需要购买硬件、不需要安装配置各种环境；数据和计算资源一直处在“在线”状态，不必担心数据太大或计算资源不足的问题。

阿里云机器学习算法平台不仅在阿里集团内部使用，也已对阿里集团外部开放，读者可以通过阿里云官网试用或使用本书中介绍的功能。

机器学习平台提供了一个舞台，主角是其上面的近百种算法。本书的重点放在这些算法的使用上——通过实际的数据和具体的场景，帮助读者理解各算法所擅长处理的问题；另外，本书是根据机器学习的知识点由浅入深来逐步组织的，以降低阅读本书的门槛，使读者对所学的内容能产生清晰的印象。

在具体章节的组织上，阿里云机器学习平台的介绍占两个章节，即第 1 章和附录 A。第 1 章为平台简介，在内容组织上尽量减少文字说明，将最基本的内容用图例来表示；附录 A 介绍了些琐碎但重要的事情，像如何试用、如何上传数据以及预处理函数的详细说明。第 2 章至第 12 章是按照机器学习的知识点逐步深入的思路来编排的。分类模型是机器学习理论和应用方面

的重头，首先是数值类型特征的二分类模型、扩展特征的类型、多分类模型；之后介绍聚类模型；然后是回归模型；再后面介绍文本分析领域的应用（主题模型、向量化、关键词等），根据文本描述进行预测、情感分析，并以电影数据为例，搭建推荐系统。深度学习的内容放在第 12 章，围绕 TensorFlow 框架组件，介绍了一个能体现 TensorFlow 特点的 Softmax 模型的例子，然后介绍了使用深度学习 DNN 分类器的例子。

机器学习平台降低了我们使用机器学习知识的门槛，将各个算法作为组件，即使不了解其背后的理论知识，读者仍然可以仿照书中实例，将组件连接起来解决一些实际问题。希望本书能帮助读者在机器学习的实践中学习。

最后，感谢一起研发阿里云机器学习平台的各位同事！感谢家人的理解和支持！

杨旭

2017 年 7 月

目录

第 1 章　阿里云机器学习 1

1.1　产品特点 1

1.2　名词解释 2

1.3　构建机器学习实验 3

1.3.1　新建实验 3

1.3.2　使用组件搭建工作流 4

1.3.3　运行实验、查看结果 5

1.3.4　模型部署、在线预测 6

第 2 章　商家作弊行为检测 7

2.1　数据探索 8

2.2　建模、预测和评估 15

2.3　尝试其他分类模型 19

2.4　判断商家作弊 24

第 3 章　生存预测 27

3.1　数据集一 27

3.1.1　特征分析 28

3.1.2　生存预测 33

3.2　数据集二 36

3.2.1　随机森林模型 39

3.2.2　朴素贝叶斯模型 47

第 4 章　信用风险预测 50
4.1　整体流程 53
4.1.1　特征哑元化 54
4.1.2　特征重要性 57
4.2　模型效果评估 61
4.3　减少模型特征的个数 62
第 5 章　用户购买行为预测 65
5.1　数据探索 66
5.2　思路 68
5.2.1　用户和品牌的各种特征 69
5.2.2　二分类模型训练 71
5.3　计算训练数据集 71
5.3.1　原始数据划分 72
5.3.2　计算特征 74
5.3.3　计算标签 89
5.4　二分类模型训练 90
5.4.1　正负样本配比 90
5.4.2　逻辑回归算法 92
5.4.3　随机森林算法 94
第 6 章　聚类与分类 96
6.1　数据可视化 97
6.2　K-Means 聚类 98
6.2.1　聚类、评估流程 100
6.2.2　聚成两类 101
6.2.3　聚成三类 103
6.3　K 最近邻算法 104
6.3.1　使用 KNN 算法进行分类 105
6.3.2　算法比较 108
6.4　多分类模型 109
6.4.1　使用朴素贝叶斯算法 109
6.4.2　使用逻辑回归多分类算法 112
6.4.3　使用随机森林算法 115
6.4.4　各多分类模型效果对比 118

第 7 章 葡萄酒品质预测 119
7.1 数据探索 120
7.2 线性回归 123
7.3 GBDT 回归 125
第 8 章 文本分析 127
8.1 分词 128
8.2 词频统计 130
8.3 单词的区分度 131
8.4 字符串比较 133
8.5 抽取关键词、关键句 139
8.5.1 原理简介 139
8.5.2 完整流程 141
8.6 主题模型 146
8.6.1 LDA 模型 147
8.6.2 新闻的主题模型 149
8.6.3 数据预处理 150
8.6.4 主题与原始分类的关系 153
8.7 单词映射为向量 160
8.7.1 相近单词 162
8.7.2 单词聚类 165
8.8 组件使用小结 168
第 9 章 基于用户退货描述的赔付预测 170
9.1 思路 171
9.2 训练集的特征生成 173
9.3 测试集的特征生成 180
9.4 模型训练、预测、评估 181
9.5 提高召回率 185
第 10 章 情感分析 189
10.1 词袋模型 190
10.1.1 训练集的特征生成 192
10.1.2 测试集的特征生成 196
10.1.3 模型训练、预测、评估 197

10.2 词向量模型.....200
10.2.1 特征生成.....201
10.2.2 模型训练.....206
第 11 章 影片推荐.....211
11.1 协同过滤.....212
11.2 整体流程.....213
11.3 预处理，过滤出好评信息.....215
11.4 计算影片间的相似度.....215
11.5 计算用户可能喜欢的影片.....221
11.6 查看推荐效果.....224
第 12 章 支持深度学习框架.....227
12.1 TensorFlow 组件简介.....227
12.2 Softmax 模型.....231
12.3 深度神经网络.....234
附录 A.....237

第1章 阿里云机器学习

阿里云机器学习平台是构建在阿里云 MaxCompute 计算平台之上，集数据处理、建模、离线预测、在线预测为一体的机器学习算法平台。用户通过拖曳可视化的操作组件来进行试验，使得没有机器学习背景的工程师也可以轻易上手玩转数据挖掘。平台提供了丰富的组件，包括数据预处理、特征工程、算法组件、预测与评估。平台目前整合了阿里集团内最先进的算法，为集团内、外不同用户提供算法服务。

欢迎访问阿里云机器学习的网址：https://data.aliyun.com/product/learn。用户可在阿里云网站申请公测，进行免费试用，相关内容详见本书附录。

1.1 产品特点

- 简单、易用

 将各个复杂的机器学习算法抽象为算法组件，通过拖曳组件的方式即可完成机器学习流程的搭建，大大降低了机器学习算法学习和使用的门槛。

- 算法丰富、完整

 不但包括了机器学习核心的分类、聚类、回归模型，还包括了数据探索、预处理、特征工程、深度学习、文本分析等方面的组件，可以一站式地完成不同场景的解决方案。

- 支持处理大数据

 提供高性能的机器学习算法实现，并根据数据量的大小及计算的复杂程度自动获取适合的计算资源，再多的数据也能及时处理。

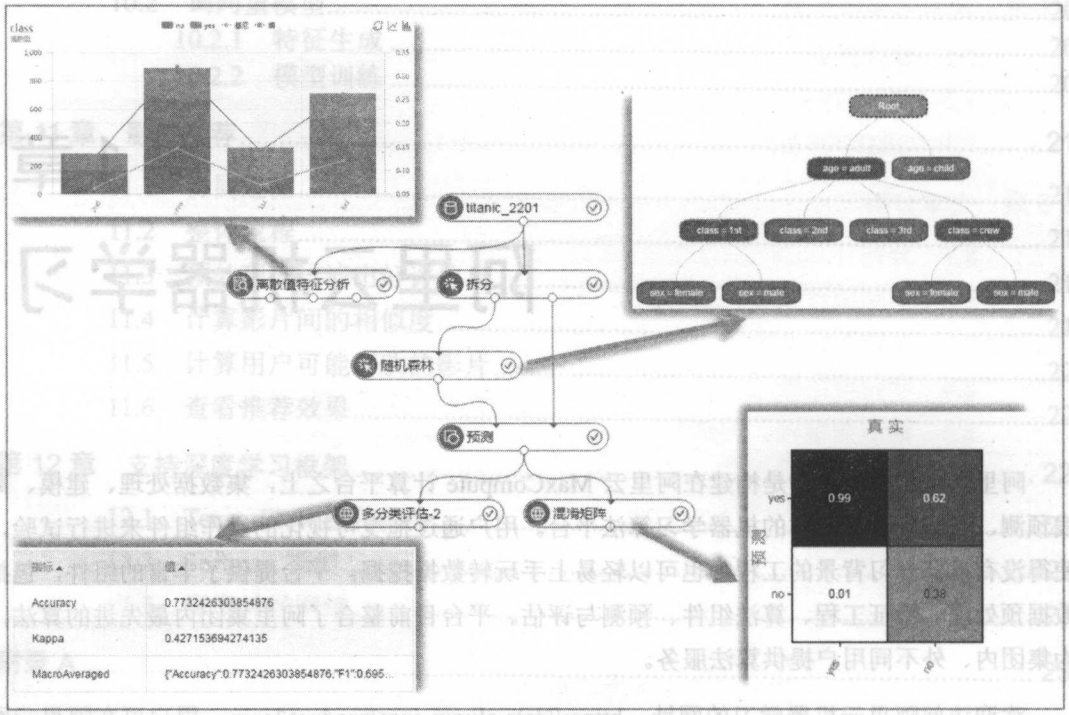


图 1-1　阿里云机器学习

1.2　名词解释

为便于读者阅读，将阿里云机器学习平台中涉及的一些名词进行了解释，详见表 1-1。

表 1-1　名词解释

名　词	解　释
MaxCompute	开放数据处理服务，由阿里云自主研发，提供针对 TB/PB 级数据、实时性要求不高的分布式处理能力，应用于数据分析、挖掘、商业智能等领域
项目（Project）	项目（也称项目空间）是 MaxCompute 最基本的组织对象。其他对象，例如表（Table）和实例（Instance）等都归属于某个项目
实验（Experiment）	实验是指阿里云机器学习平台用户搭建的数据工作流程或者数据应用。用户需要先建立一个实验实例，然后在实验画布上搭建数据流程

续表

名　词	解　释
MaxCompute 源表与 MaxCompute 目标表（Table）	表（Table）是 MaxCompute 中数据存储对象。与常见的关系型数据类似，MaxCompute 中的表逻辑上也是二维结构。源表指一个算法节点的输入，目标表指算法节点的输出
组件（Nodes）	组件是用户可以在阿里云机器学习平台上调用执行的最小操作单元，例如数据导入导出、数据处理、数据分析、模型训练或者预测
模型（Model）	模型是特指一个算法或者机器学习训练组件产生的结果数据。模型是一类特殊的组件
分区（Partition）	MaxCompute 表分区

1.3　构建机器学习实验

1.3.1　新建实验

如图 1-2 所示，点击左侧“实验”按钮，右击“我的实验”选项，选择“新建空白实验”或“从模板新建实验”选项，然后系统会自动进入新建的实验操作空间。

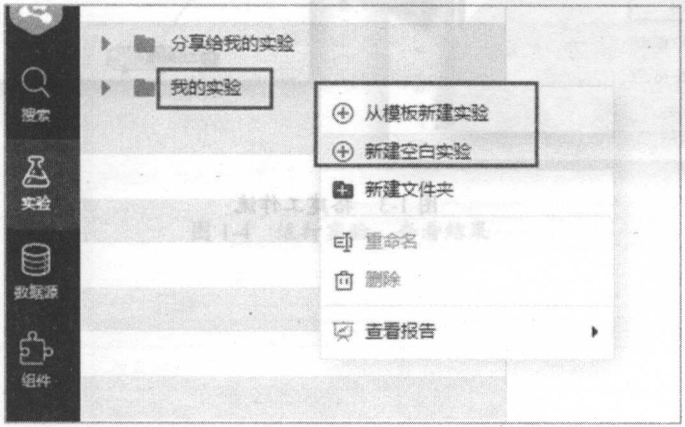


图 1-2　新建实验

1.3.2 使用组件搭建工作流

如图 1-3 所示，拉入数据表和算法组件，进行实验流搭建。具体操作是：点击左侧“数据源”按钮搜索选择需要的数据表，拖曳到右侧空白处；点击左侧“组件”按钮，选择需要的组件，并拖曳到右侧空白处；并根据实验流程，连接组件的输入、输出桩。

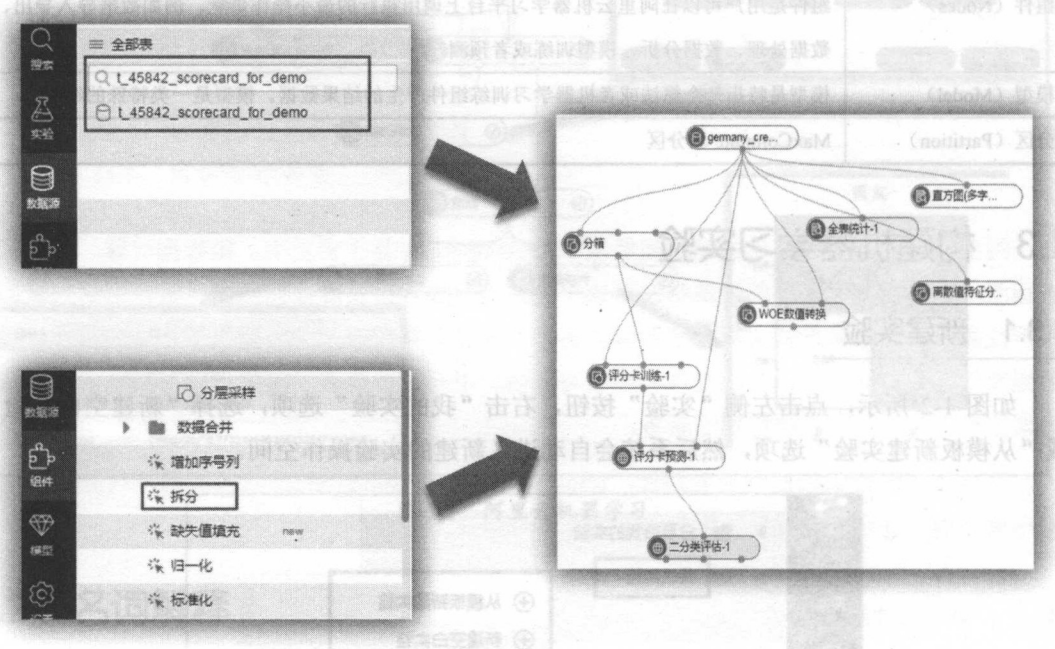


图 1-3 搭建工作流

1.3.3　运行实验、查看结果

如图 1-4 所示，点击工作区下方的“运行”按钮，依次运行实验的各个组件，组件运行完成后，其右端会显示绿色的对号标记，然后，单击鼠标右键，就可选择查看结果数据及图表。

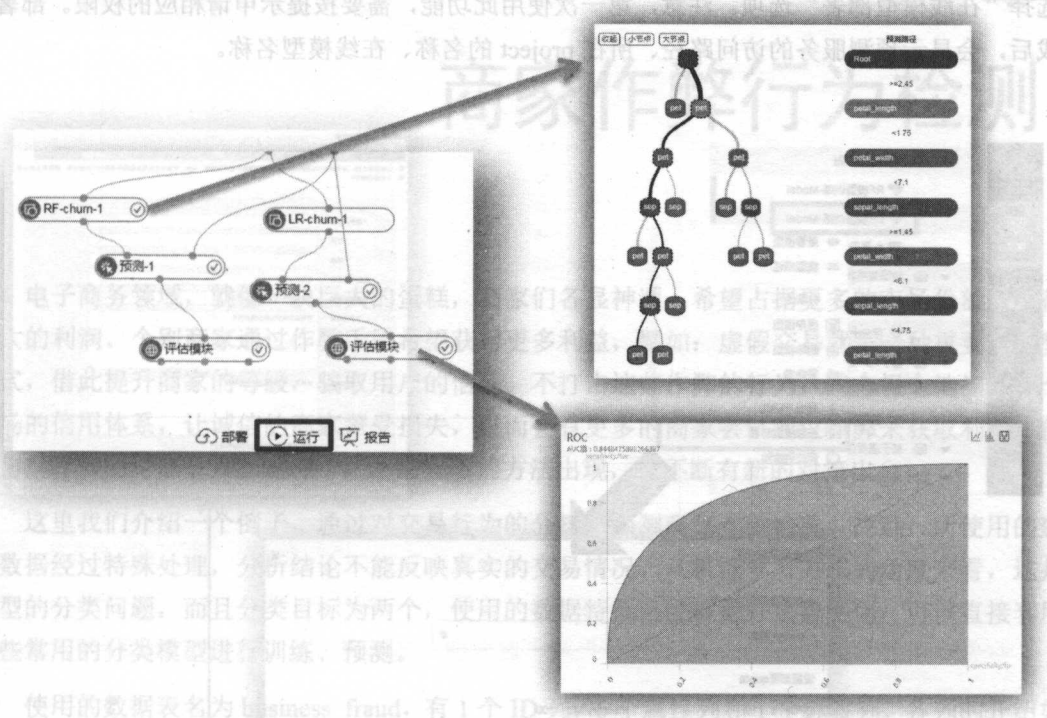


图 1-4　运行实验、查看结果

1.3.4 模型部署、在线预测

1）模型部署。

如图 1-5 所示，点击左侧“模型”按钮，找到当前实验名称，选择模型，然后在右键菜单选择“在线模型部署”选项。注意，第一次使用此功能，需要按提示申请相应的权限。部署完成后，会显示预测服务的访问路径、所在 project 的名称、在线模型名称。

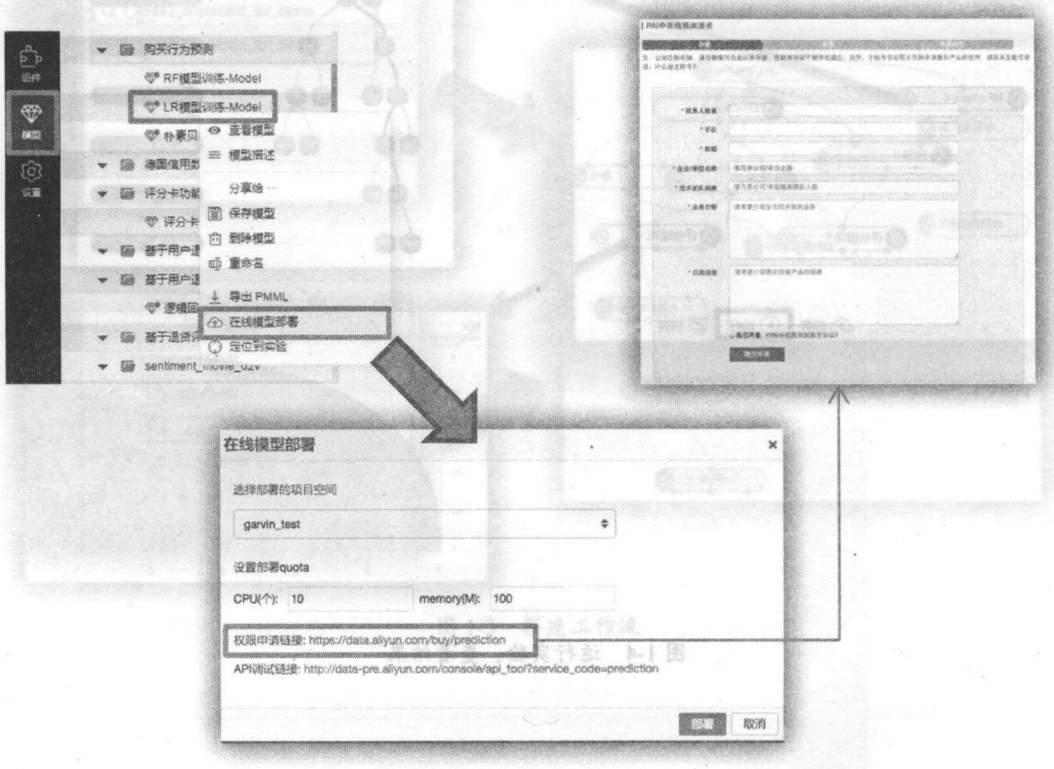


图 1-5　在线模型部署

2）在线预测。

预测 API 请求地址：$访问路径/projects/$project 名称/onlinemodels/$模型名称。

请求 Body 内容需要填上对应算法所输入的 json 格式文件，包括字段名、字段的 type 和具体数值。

第2章 商家作弊行为检测

电子商务领域，就像一块巨大的蛋糕，商家们各显神通，希望占据更多的市场份额，获得更大的利润。个别商家通过作弊手段希望获得更多利益，譬如：虚假交易就是一种重要的作弊方式，借此提升商家的等级，骗取用户的信任。不打击这些作弊的行为，就会极大地损害整个市场的信用体系，让诚信的商家蒙受损失，进而会有更多的商家尝试通过作弊来获取利益。作弊与反作弊的斗争一直在进行中，不断有新的方法出现，也不断有新的对策出台。

这里我们介绍一个例子，通过对交易行为的分析，预测商家作弊情况。注意：所使用的建模数据经过特殊处理，分析结论不能反映真实的交易情况。从机器学习方法的角度来看，这是典型的分类问题，而且分类目标为两个，使用的数据特征已经被很好地数字化，可以直接套用一些常用的分类模型进行训练、预测。

使用的数据表名为 business_fraud，有 1 个 ID 列，6 个属性列和 1 个标签列，各列的介绍如表 2-1 所示，各属性列如何变换到 0~1 区间，不是本节的重点，不展开讨论。

表 2-1　交易信息字段

列　名	含　义	备　注
Transaction_id	交易 ID	
a_score	账户价值	取值范围：[0，1]
b_score	购物类别	取值范围：[0，1]
r_score	消费频率	取值范围：[0，1]
p_score	注册时间	取值范围：[0，1]
ri_score	购物时间	取值范围：[0，1]
v_score	消费总额	
label	是否作弊	0-作弊操作，1-非作弊

数据如图 2-1 所示，很明显，字段 b_score、r_score 和 p_score 中 0 值的个数较多。

transaction_id	a_score	b_score	r_score	p_score	ri_score	v_score	label
0	0.177	0	0	0	0.373	0.019	0
1	0.002	0.919	0	0.311	0.517	0.329	0
2	0.362	0	0	0	0.195	0.02	0
3	0	0	0	0	0.37	0.176	0
4	0.808	0	0	0	0.236	0.667	0
5	0.002	0	0	0	0.462	0	0
6	0.718	0	0	0	0.152	0.422	0
7	0.818	0	0	0	0.174	0.565	0
8	0.711	0	0	0	0.118	0.279	0
9	0.01	0	0	0	0.279	0.199	0
10	0.709	0	0	0	0.259	0.34	0
11	0.135	0	0	0	0.253	0.096	0
12	0.848	0	0	0	0.093	0.705	0
13	0.012	0	0	0	0.279	0	0
14	0.085	0	0	0	0.512	0.002	0
15	0.189	0	0	0	0.419	0.007	0
16	0.847	0	0	0	0.066	0.456	0

图 2-1　交易信息数据表

2.1　数据探索

首先使用最常用的组件，“全表统计”和“直方图（多字段）”，关注最基本的统计信息。各组件的连接方式如图 2-2 所示。

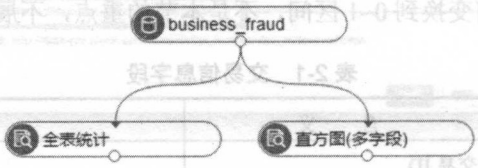


图 2-2　数据探索流程图

其中，“全表统计”组件可以使用默认参数，即对所有数据列进行统计；“直方图（多字段）”组件需要选择字段，如图 2-3 所示，在弹出的“选择字段”窗口选择所有特征属性列。

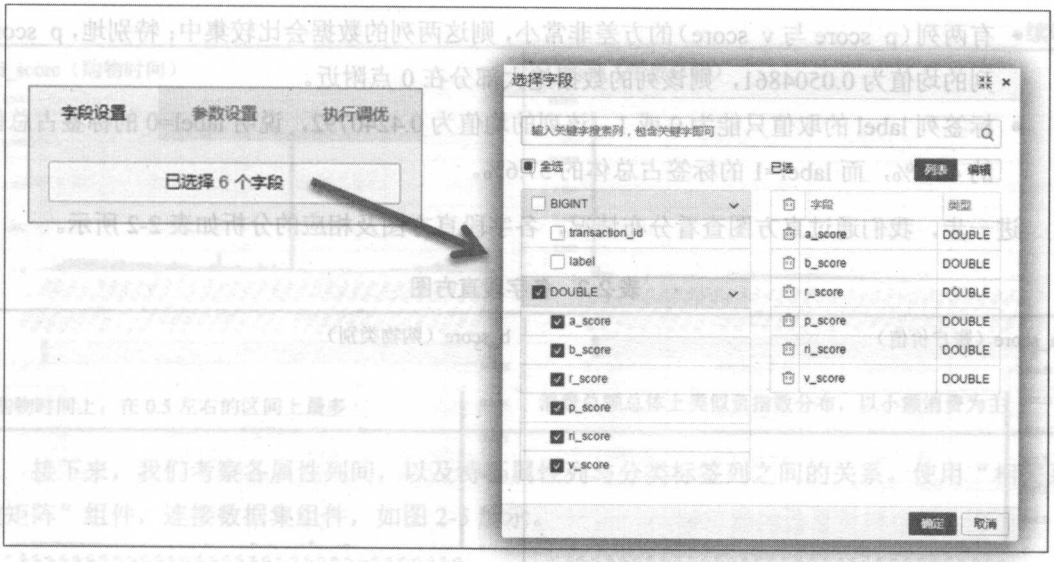


图 2-3　直方图组件设置

“全表统计”的计算结果如图 2-4 所示，每个字段的统计值占一行，第一列是字段名称，随后是各种统计指标。

数据探查 - pai_temp_6969_112916_1 - (仅显示前一百条)

colname	datatype	totalcount	count	missingcount	nancount	positiveinfinitycount	negativeinfinitycount	min	max	mean	variance
a_score	double	34727	34727	0	0	0	0	0	0.973	0.3308785...	0.1081052...
b_score	double	34727	34727	0	0	0	0	0	1	0.2611760...	0.1318923...
label	bigint	34727	34727	0	0	0	0	0	1	0.4240792...	0.2442430...
p_score	double	34727	34727	0	0	0	0	0	1	0.0504861...	0.0355333...
r_score	double	34727	34727	0	0	0	0	0	1	0.2817368...	0.1878804...
ri_score	double	34727	34727	0	0	0	0	0.021	0.908	0.4625686...	0.0726257...
transactio...	bigint	34727	34727	0	0	0	0	0	34726	17363	100499938
v_score	double	34727	34727	0	0	0	0	0	0.955	0.2082491...	0.0537965...

关闭

图 2-4　全表统计结果

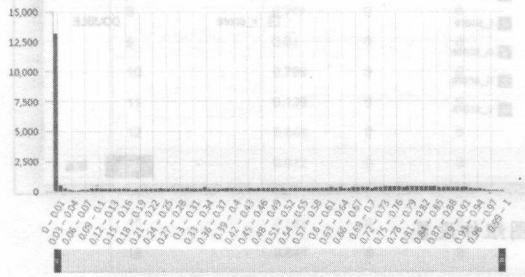
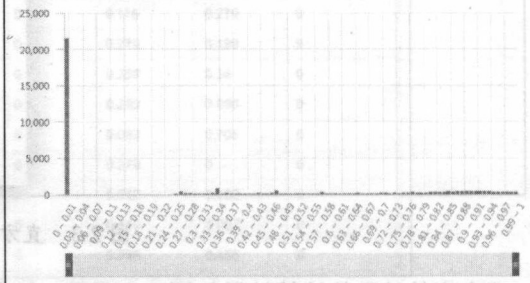
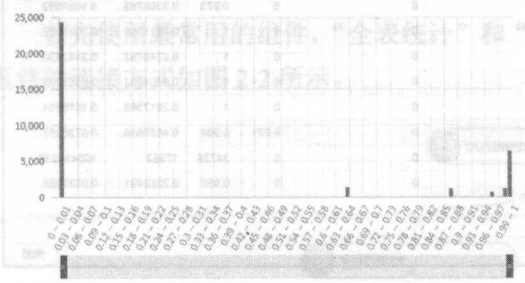
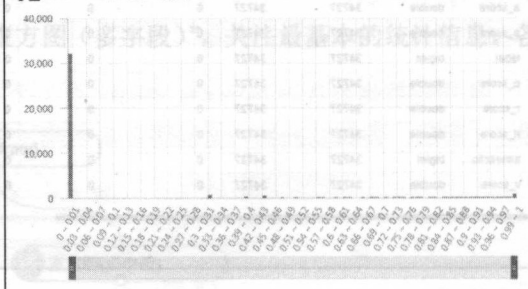
由图 2-4 的统计结果截图可以看出：

- 数据集的样本数量为 34727，各列没有缺失值。
- 6 个属性列的取值都在 0~1 之间。
- a_score 列、ri_score 列和 v_score 列的最大值没有达到 1；ri_score 列的最小值为 0.021，略大于 0。

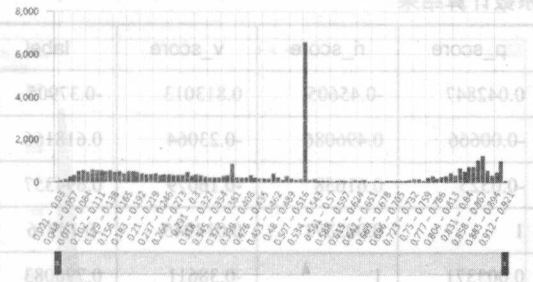
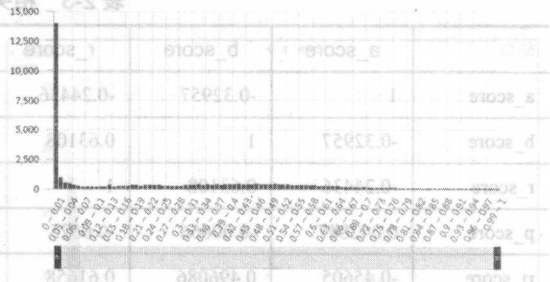
- 有两列（p_score 与 v_score）的方差非常小，则这两列的数据会比较集中；特别地，p_score 列的均值为 0.0504861，则该列的数据绝大部分在 0 点附近。
- 标签列 label 的取值只能为 0 或 1，该列的均值为 0.4240792，说明 label=0 的标签占总体的 42.4%，而 label=1 的标签占总体的 57.6%。

进一步，我们通过直方图查看分布情况，各字段直方图及相应的分析如表 2-2 所示。

表 2-2　各字段直方图

a_score（账户价值）  在 0 点附近，即很低价值的账户占了很大的比重，约有 13000+个，约占总数的 40%；其他区间的用户分布比较平均	b_score（购物类别）  购物类别非常少的有 21000+个，在总体中占比达到 60~70%
r_score（消费频率）  出现非常明显的两级分化情况，消费频率很低的有 24000+，约占总数的 70%；而消费频率非常高的有 6000+，占到总数的将近 20%	p_score（注册时间）  绝大部份都是在开始的时候注册，个数为 31000+，约占总数的 90%

续表

ri_score（购物时间）	v_score（消费总额）
	
购物时间上，在 0.5 左右的区间上最多	消费总额总体上类似负指数分布，以小额消费为主

接下来，我们考察各属性列间，以及特征属性列与分类标签列之间的关系。使用“相关系数矩阵”组件，连接数据集组件，如图 2-5 所示。

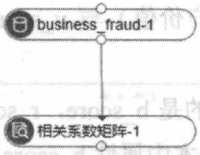


图 2-5　计算相关系数矩阵

设置“相关系数矩阵”组件的参数，在“字段设置”中选择所有属性列，它们都是双精度 DOUBLE 数据类型的，并选择 label 列（BIGINT 数据类型），如图 2-6 所示。

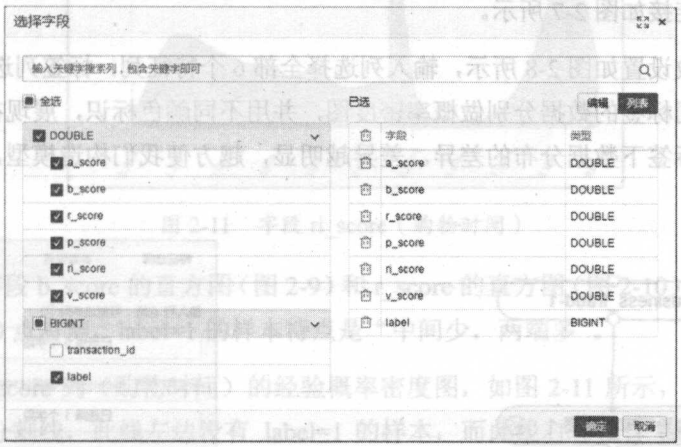


图 2-6　选择计算相关系数的字段

计算结果如表 2-3 所示。

表 2-3　相关系数计算结果

	a_score	b_score	r_score	p_score	ri_score	v_score	label
a_score	1	-0.32957	-0.24436	0.042847	-0.45605	0.813013	-0.37905
b_score	-0.32957	1	0.63108	-0.00666	0.496086	-0.23064	0.618166
r_score	-0.24436	0.63108	1	-0.0525	0.61658	-0.16079	0.694357
p_score	0.042847	-0.00666	-0.0525	1	0.003371	0.014945	-0.14976
ri_score	-0.45605	0.496086	0.61658	0.003371	1	-0.38611	0.790083
v_score	0.813013	-0.23064	-0.16079	0.014945	-0.38611	1	-0.30304
label	-0.37905	0.618166	0.694357	-0.14976	0.790083	-0.30304	1

由表 2-3 中相关系数的值可知：

- 相关度最高的是列 a_score（账户价值）与 v_score（消费总额），相关系数为 0.813013，为正相关
- 可以看到与 label 列相关度较高的是 b_score，r_score 与 ri_score，且相关系数都为正数，说明是正相关。即，label=0 的样本中属性 b_score，r_score 与 ri_score 会偏小；而 label=1 的样本中属性 b_score，r_score 与 ri_score 会偏大。

有了上面的分析，我们对数据有了一定的了解，但是对于我们关心的分类问题还是不够的。下面我们将根据标签列 label 取值的不同，对比观察各属性列分布的差异。选择“经验概率密度图”组件，组件连接如图 2-7 所示。

该组件的参数设置如图 2-8 所示，输入列选择全部 6 个特征列，标签列选择“label”，这样可以对分属于不同标签的数据分别做概率密度图，并用不同颜色标识，展现在同一个图中，便于我们发现不同标签下数据分布的差异。差异越明显，越方便我们构造模型，对不同的标签进行区分。

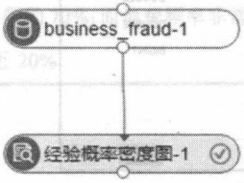


图 2-7　计算经验概率密度图

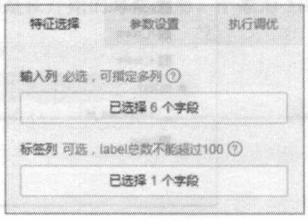


图 2-8　经验概率密度图组件的参数设置

查看分析报告，可以看到属性列 b_score，r_score 与 ri_score 对应不同的标签值时，经验概率密度图差异较大，如图 2-9、图 2-10 和图 2-11 所示。

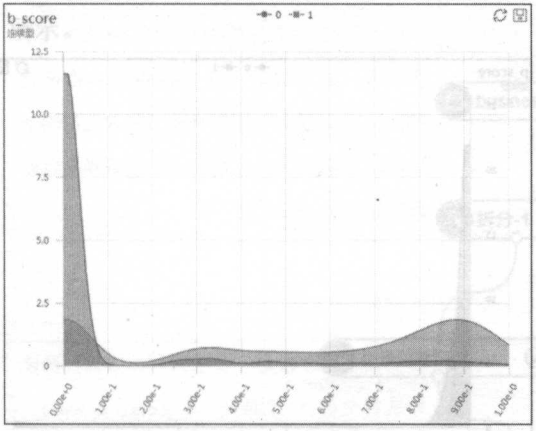


图 2-9　字段 b_score（购物类别）

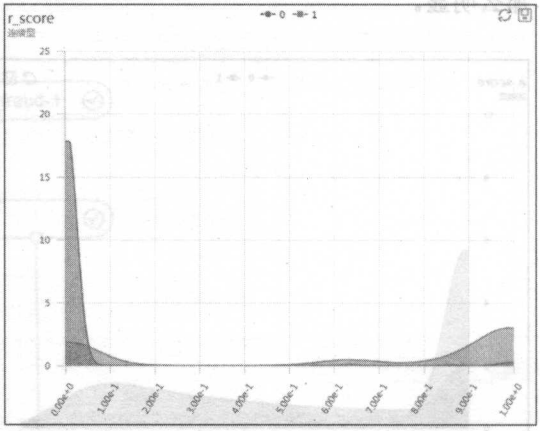


图 2-10　字段 r_score（消费频率）

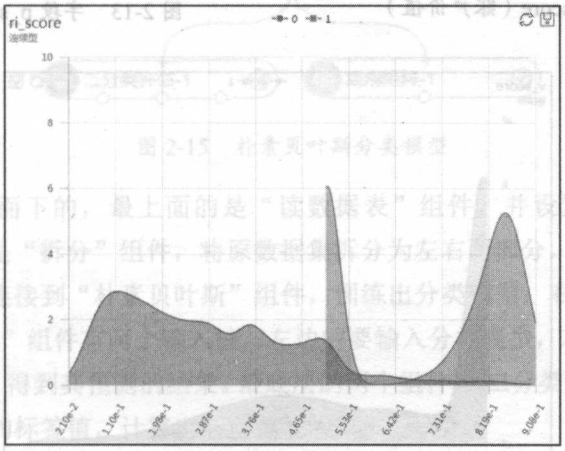


图 2-11　字段 ri_score（购物时间）

容易看出，字段 b_score 的直方图（图 2-9）和 r_score 的直方图（图 2-10）比较相似，label=0 的样本都集中在 0 点右侧，label=1 的样本特点是“中间少，两端多”。

我们再看 ri_score 列（购物时间）的经验概率密度图，如图 2-11 所示，可以看到在 0.5 附近有一条明显的分割线，此线左边没有 label=1 的样本，而此线右边包含 label=0 的样本较少，且概率密度迅速下降。这三个图也形象地解释了为什么 b_score，r_score 以及 ri_score 与 label

列的相关系数值较大。

其他 3 个属性的经验概率密度图如图 2-12 、图 2-13 和图 2-14 所示，两条曲线间的差异没那么明显。

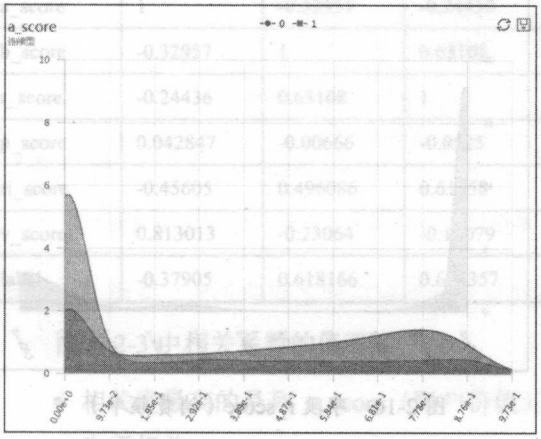


图 2-12　字段 a_score（账户价值）

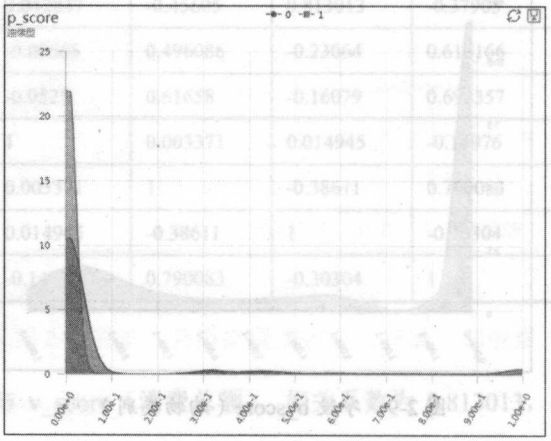


图 2-13　字段 p_score（注册时间）

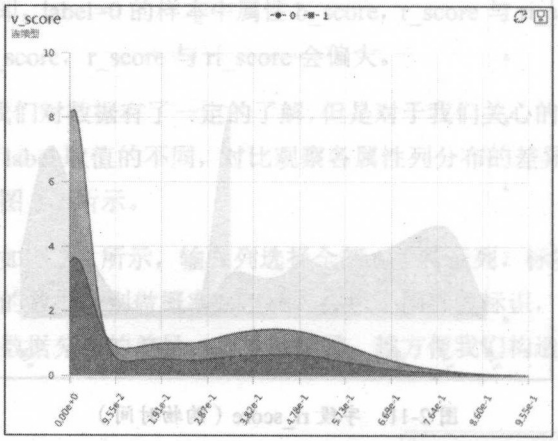


图 2-14　字段 v_score（消费总额）

2.2　建模、预测和评估

我们以朴素贝叶斯分类模型为例，介绍一般的建模、预测和评估流程。整体流程如图 2-15 所示。

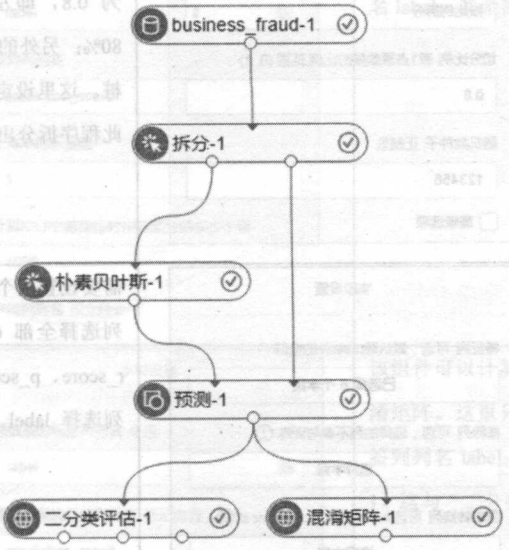


图 2-15　朴素贝叶斯分类模型

整个流程是自上而下的，最上面的是“读数据表”组件，并设置读取的数据表名称为 business_fraud。随后是“拆分”组件，将原数据集拆分为左右两部分，左边输出的数据集会被用作模型训练，将其连接到“朴素贝叶斯”组件，训练出分类模型；右边输出的数据集用来检验模型的效果。“预测”组件有两个输入桩，左边的要输入分类模型，右边的要输入测试数据，根据测试数据的特征，得到其预测的结果。最底层的两个组件（“二分类评估”和“混淆矩阵”），会由预测结果及原始的标签值，计算出模型评估的结果。

下面逐个介绍组件详细的参数配置，如表 2-4 所示，第一列为所用的组件，第二列为相应的参数配置截图，第三列为详细的说明。

表 2-4　各组件参数配置

组　件	参数设置	说　明
拆分-1	参数设置　执行调优 拆分方式 按比例拆分 切分比例 表1占原数据的比例,范围 (0, 1) 0.8 随机数种子 正整数 123456 高级选项	拆分方式可以选择：“按数据拆分”或“按比例拆分”。这里选择按比例拆分，并设定比例为 0.8，即左边输出桩的数据占总数据的 80%；另外的 20%数据会输出到右边的输出桩。这里设定了随机数种子，保证两次运行此程序拆分出的结果一致
朴素贝叶斯-1	字段设置 特征列 可选，默认除label外的全部 已选择 6 个字段 排除列 可选，选择的列不参与训练 选择字段 强制转换列 可选,bigint解析为categorical 选择字段 标签列 label	需要设置两个参数：特征列和标签列。特征列选择全部 6 个特征：a_score、b_score、r_score、p_score、ri_score 和 v_score。标签列选择 label
预测-1	特征列 默认全选 选择字段 原样输出列 推荐添加label列,方便评估 已选择 8 个字段 输出结果列名 prediction_result 输出分数列名 prediction_score 输出详细列名 prediction_detail 稀疏矩阵 格式: k1:v1, k2:v2	基本上都可以使用默认参数。在“原样输出列”制定样本中需要保存的列，这里选择了 6 个属性列，1 个标签列，还有 id 列

续表

组　件	参数设置	说　明
二分类评估-1	字段设置 原始标签列列名 label 分数列列名 prediction_score 正样本的标签值 1 计算KS,PR等指标时按等频分成多少个桶 1000 分组列列名 仅支持str	该组件主要是绘制出各种评估图标，并计算出 AUC、F1 等指标。需要指定原始标签列列名 label；其余的参数使用默认值即可
混淆矩阵-1	字段设置 原数据的标签列列名 必选 label 预测结果的标签列列名 若无阈值，必选 prediction_result 阈值 可选，大于该阈值的样本作为正样本 预测结果的详细列列名 若指定阈值，必选 与标签列列名不能共存 正样本的标签值 若无阈值，可选；否则，必选 1	该组件可以计算出真实值与预测值之间的混淆矩阵。这里只设置了两个参数原数据的标签列列名 label；并设置了正样本的标签值为 1，这与“二分类评估”组件的设置是一致的

运行结束后，我们可以从两个评估组件查看模型效果。首先看比较容易理解的“混淆矩阵”组件，右键选择“查看评估报告”，可以显示不同 label 样本的预测个数及比例。如图 2-16 所示的混淆矩阵，可以看到 label 值为 0 预测仍为 0 的比例是 0.94；如图 2-17 所示的比例矩阵，label 值为 1 预测仍为 1 的比例也是 0.94。



图 2-16　混淆矩阵

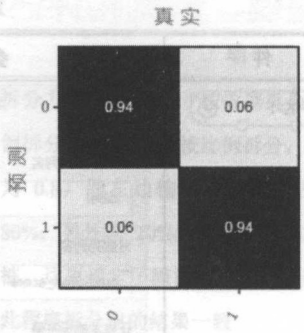


图 2-17　比例矩阵

随后，我们查看“二分类评估”组件的结果。其中涉及初学者比较陌生的两个概念：ROC 曲线和 AUC 值。我们先简单介绍一下。

ROC（Receiver Operating Characteristic）曲线最初用来评价雷达性能，被称为接收者操作特性曲线。ROC 曲线是根据一系列不同的阈值，得到相应的一系列二分类预测方式，每一个预测结果以一个点表示，每个点以真阳性率（TPR）为纵坐标，以假阳性率（FPR）为横坐标，再将所有的点连接起来，绘制成曲线。真阳性率代表获利能力，其值越高，获得的利益越多。假阳性率可以看作是成本，其值越高，成本越高。ROC 曲线中斜率较高的一段获得较大的利益，同时付出较小的成本。

ROC 曲线的范围是由（0,1）、（1,1）、（1,0）、（0,0）四点构成的单位正方形区域。最好的可能预测方式在左上角，即（0,1）点，这代表 100%灵敏（没有假阴性）和 100%特异（没有假阳性），而（0,1）点被称为完美分类器。一个完全随机预测会得到一条从左下到右上对角线（也叫无识别率线）上的一个点。一个最直观的随机预测的做决定的例子就是抛硬币。这条斜线将 ROC 空间划分为两个区域，在这条线以上的点代表一个好的分类结果，而在这条线以下的点代表一个差的分类结果。

为了将 ROC 曲线概括成单一的数量，可以选用曲线下方的面积（Area Under Curve）。一般来说，面积越大，分类预测效果越好。

“二分类评估”组件显示当前模型的 ROC 曲线，如图 2-18 所示，AUC 值为 0.9797，效果还是不错的。

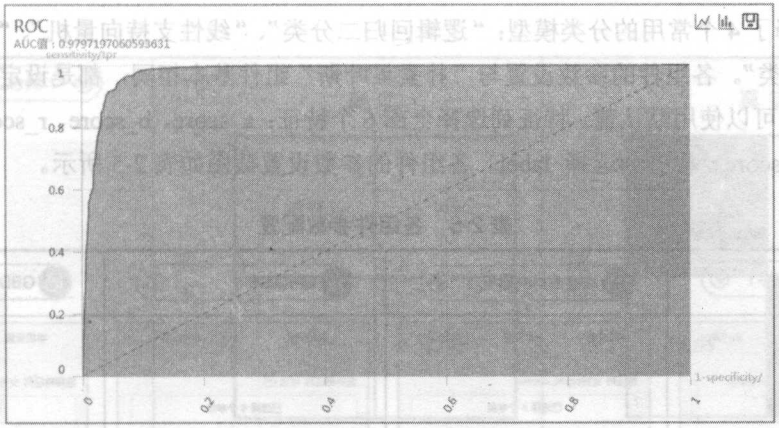


图 2-18　ROC 曲线图

2.3　尝试其他分类模型

若要尝试其他分类模型，我们只要将图 2-15 所示流程图中的“朴素贝叶斯”组件替换为其他的分类模型组件，并进行相应的参数配置即可，图 2-19 显示了替换为“逻辑回归二分类”组件的实验过程。

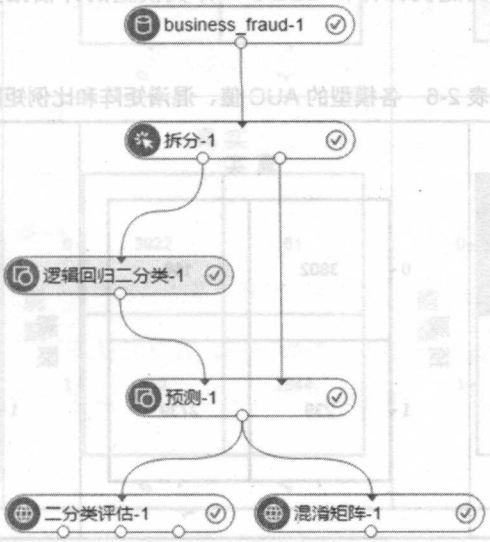
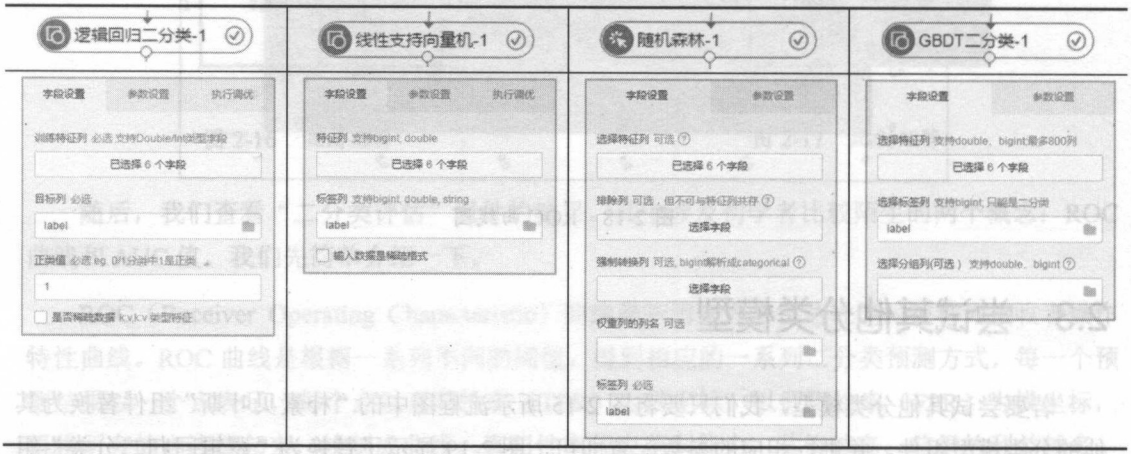


图 2-19　逻辑回归二分类模型

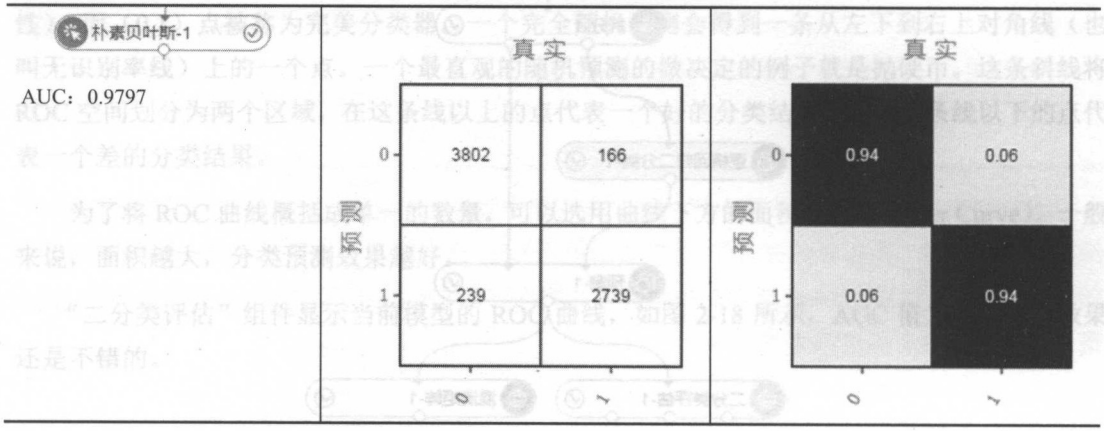
我们选择了 4 个常用的分类模型：“逻辑回归二分类”、“线性支持向量机”、“随机森林”和“GBDT 二分类”。各组件的参数设置与“朴素贝叶斯”组件基本相同，都是设定特征列和标签列，其他参数可以使用默认值。特征列选择全部 6 个特征：a_score、b_score、r_score、p_score、ri_score 和 v_score。标签列选择 label。各组件的参数设置截图如表 2-5 所示。

表 2-5　各组件参数配置

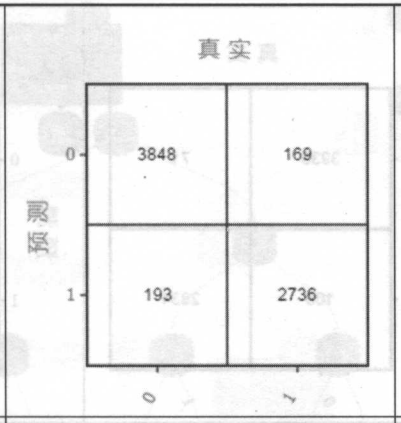
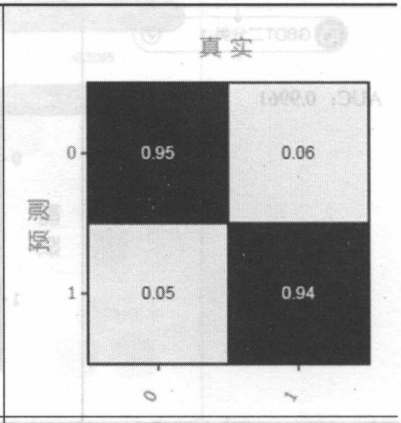
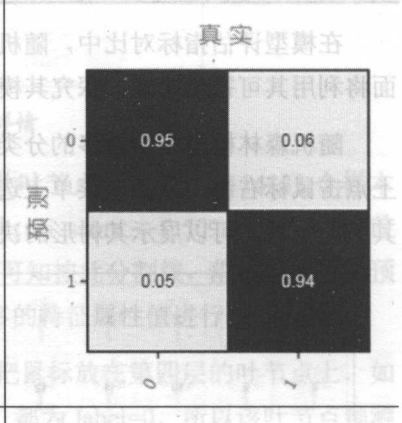
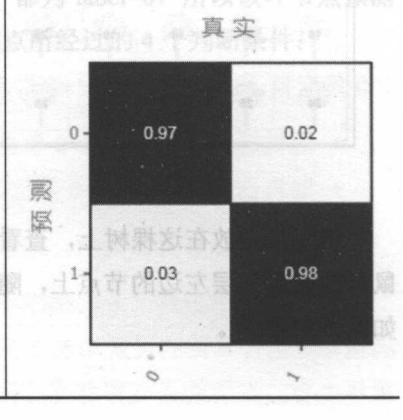


各种模型的评估结果汇总如表 2-6 所示。可以看到，逻辑回归模型和线性支持向量机模型略好于朴素贝叶斯模型，而随机森林和 GBDT 二分类模型的评估指标上了更高的档次，明显优于前三种模型。

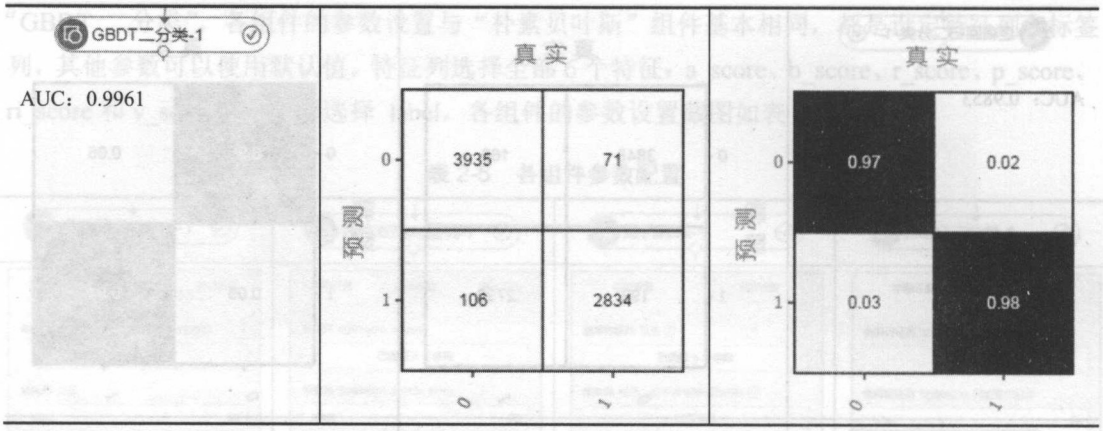
表 2-6　各模型的 AUC 值、混淆矩阵和比例矩阵



续表

逻辑回归二分类-1 AUC：0.9853		
线性支持向量机-1 AUC：0.9851		
随机森林-1 AUC：0.9957		

续表



在模型评估指标对比中，随机森林模型表现很好，而且随机森林模型便于可视化展示，下面将利用其可视化功能，探究其模型内部。

随机森林模型是由众多的分类树构成的，我们可以查看每个分类树，在“随机森林”组件上点击鼠标右键，从弹出菜单上选择“模型选项→查看模型”选项，会列出所有决策树，选中其中的一棵，可以展示其树形的决策判断过程，如图 2-20 所示。

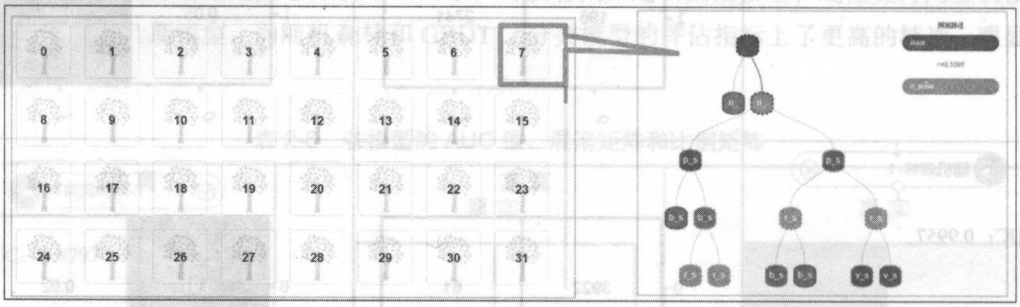


图 2-20　选择、显示决策树

将注意力放在这棵树上，查看各个判断节点。根节点是根据 ri_score 的值进行判断的，将鼠标放在第一层左边的节点上，随即会出现一个提示框，显示该节点的一些细节，得到的结果如图 2-21 所示。

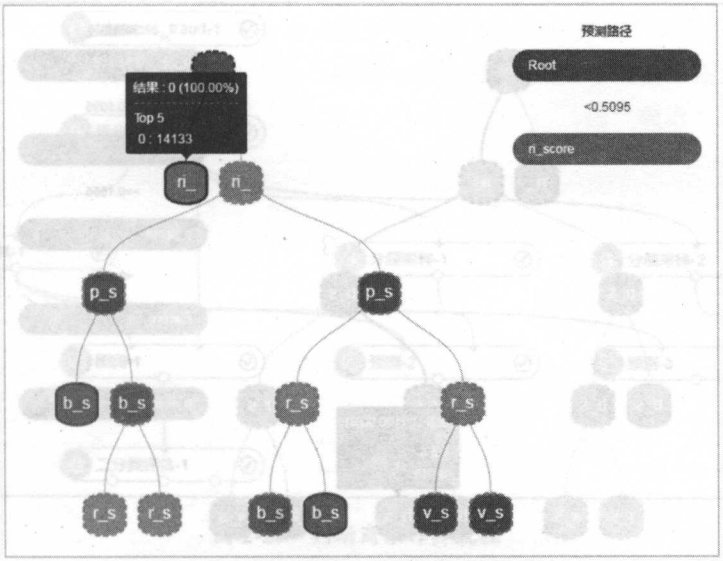


图 2-21 查看决策树第一层节点详情

该节点没有子节点，是叶节点，弹出的提示窗体显示，此叶节点在训练时有 14133 个样本落在该节点，且标签值都为 0。右侧表示到该节点的判断条件：ri_score < 0.5095。我们结合前面计算出的 ri_score 列的经验概率密度图（如图 2-11 所示），可知按此分割线，落在左端可以预测出标签为 0，落在右端，则两种可能都存在，还需引入更多的特征属性值进行判断。

第一层右边的节点会产生很多节点用于详细判断，我们把鼠标放在第四层的叶节点上，如图 2-22 所示，可以看到训练时，落在该节点的样本数为 18，都为 label=0，所以该叶节点预测时的输出结果为 0。另外，图 2-22 的右边还列出了到此叶节点所经过的 4 个判断条件：

- ri_score>=0.5095
- p_score>=0.1555
- r_score<0.316
- b_score>=0.9315

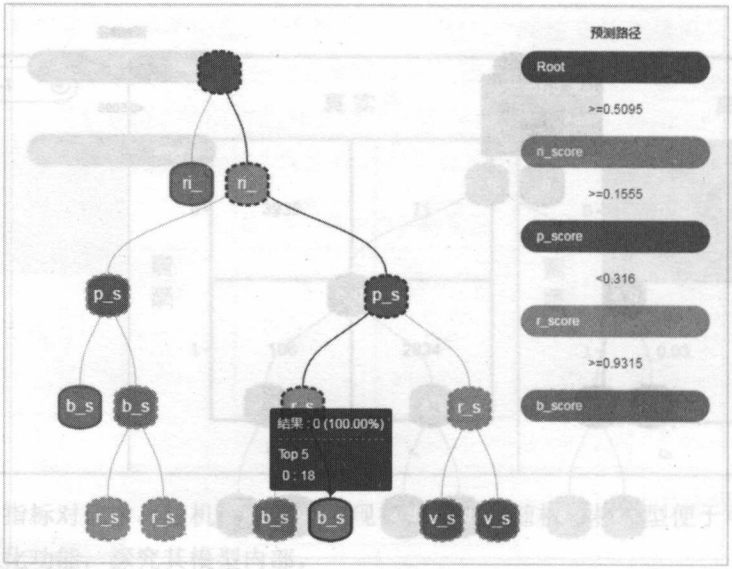


图 2-22　查看决策树第四层节点详情

2.4　判断商家作弊

前面的几个小节讲述的是，我们可以建立多种分类模型，对每个交易是否“作弊”进行判断。这一节，我们回到本章最开始的问题，如何判断商家是否作弊，也就是对该商家所进行的每个交易的“作弊”情况进行汇总，给出判断。

由于原始数据没有表示商家的字段，需要构造出两种典型商家的数据。假设每个商家都有100 条交易，由数据字段 label 的含义，0 为作弊操作，1 为非作弊，则：

- 随机选择 95 条 label=0 数据、5 条 label=1 数据作为一个作弊商家。
- 随机选择 5 条 label=0 数据、95 条 label=1 数据作为一个非作弊商家。

以随机森林分类器为例，则整体的流程图如图 2-23 所示。

其中的两个组件“分层采样-1”与“分层采样-2”，分别随机生成了作弊商家、非作弊商家的情形，组件具体的设置如表 2-7 所示，都需要设置分组列 label。然后设置每个 label 值对应的采样个数，左边的是作弊商家，选 95 个标签为 0 的，5 个标签为 1 的；右边的是非作弊商家，选 5 个标签为 0 的，95 个标签为 1 的。最后两个组件都设置上随机种子值，使得在数据不变的情况下，重复运行该组件，产生的结果是一样的，保证实验的可重复性。

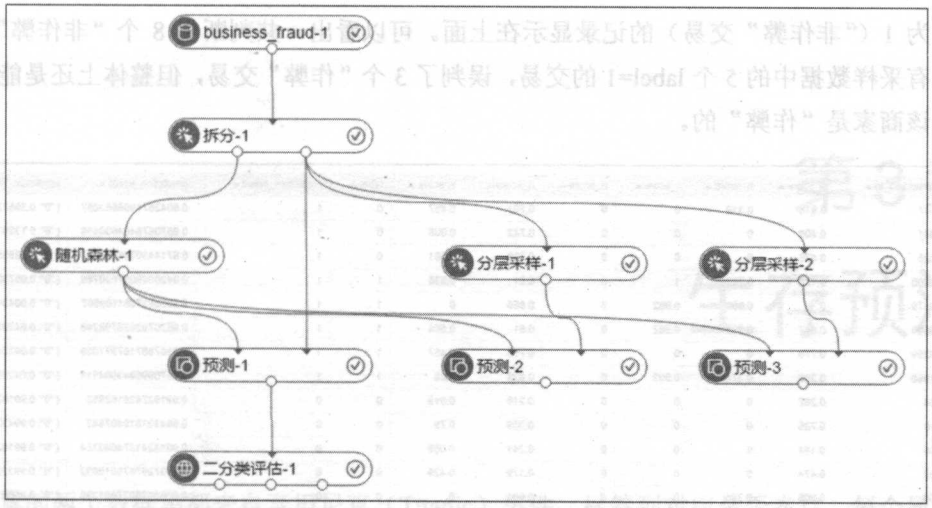


图 2-23 判断商家作弊流程

表 2-7 各分层采样组件配置对比

分层采样-1	分层采样-2
字段设置 参数设置 执行调优 分组列 必选 分层按此列划分 label	字段设置 参数设置 执行调优 分组列 必选 分层按此列划分 label
字段设置 **参数设置** 执行调优 采样个数 与采样比例二选一 0:95,1:5 采样比例 范围(0,1) 与采样个数二选一 随机种子值 可选 1234567	字段设置 **参数设置** 执行调优 采样个数 与采样比例二选一 0:5,1:95 采样比例 范围(0,1) 与采样个数二选一 随机种子值 可选 1234567

最后，我们查看每个商家对应数据集的预测结果，看看是否能判断出商家的“作弊”情况。首先，查看左边的“预测-2”组件的结果，如图 2-24 所示，按 prediction_result 降序排列，把所

有预测为 1（“非作弊”交易）的记录显示在上面。可以看出一共判断出 8 个“非作弊”交易，包括所有采样数据中的 5 个 label=1 的交易，误判了 3 个“作弊”交易，但整体上还是能明显地判断出该商家是“作弊”的。

transaction_id	a_score	b_score	r_score	p_score	ri_score	v_score	label	prediction_result	prediction_score	prediction_detail
5547	0.818	0.319	0	0	0.701	0.467	0	1	0.6042674906663087	{"0": 0.3957325(
6041	0.409	0	0	0	0.742	0.006	0	1	0.8670878464639458	{"0": 0.1329121(
9025	0.475	0	0	0	0.738	0.001	0	1	0.8714430746965417	{"0": 0.1285569:
20800	0.124	0.782	1	0	0.51	0.626	1	1	0.9426949057020769	{"0": 0.0573050(
29379	0	0.593	0.982	0	0.855	0	1	1	0.9956817061169667	{"0": 0.0043182(
29600	0.847	0.353	0.982	0	0.81	0.504	1	1	0.9520792835798246	{"0": 0.0479207(
30259	0.776	0	0	0	0.755	0.457	1	1	0.9167567187977059	{"0": 0.0832432(
31980	0.388	0.374	0.993	0	0.839	0.25	1	1	0.9570996943004318	{"0": 0.0429003(
104	0.287	0	0	0	0.216	0.019	0	0	0.991927425162853	{"0": 0.9919274:
110	0.725	0	0	0	0.329	0.79	0	0	0.994331819407542	{"0": 0.9943318:
155	0.184	0	0	0	0.241	0.069	0	0	0.9918241274083254	{"0": 0.9918241:
432	0.474	0	0	0	0.179	0.429	0	0	0.9937297875018032	{"0": 0.9937297
558	0.008	0.331	0	0	0.669	0	0	0	0.8068125977601338	{"0": 0.8068125
1314	0.682	0	0	0	0.134	0.346	0	0	0.9937297875018032	{"0": 0.9937297
1366	0.756	0	0	0.951	0.245	0.118	0	0	0.9980633053684168	{"0": 0.9980633(
1762	0.438	0	0	0	0.178	0.1	0	0	0.9944238952911928	{"0": 0.9944238(
1841	0.521	0	0	0	0.172	0.144	0	0	0.9937297875018032	{"0": 0.9937297

图 2-24 “预测-2”组件的结果

随后，查看右边的“预测-3”组件的结果，如图 2-25 所示，按 prediction_result 升序排列，把所有预测为 0（“作弊”交易）的记录显示在上面。可以看出一共判断出 7 个“作弊”交易，包括所有采样数据中的 5 个 label=0 的交易，误判了 2 个“非作弊”交易，但整体上还是能明显地判断出该商家是“非作弊”的。

transaction_id	a_score	b_score	r_score	p_score	ri_score	v_score	label	prediction_result	prediction_score	prediction_detail
1762	0.438	0	0	0	0.178	0.1	0	0	0.9944238952911928	{"0": 0.9944238(
4437	0.745	0	0	0.311	0.162	0.449	0	0	0.997610510866238	{"0": 0.9976105:
6359	0.659	0.251	0	0	0.198	0.172	0	0	0.9889056310325272	{"0": 0.9889056:
9125	0	0	0	0	0.378	0	0	0	0.9501754316057232	{"0": 0.9501754:
19191	0.288	0.904	0	0	0.447	0.158	0	0	0.9831991978894262	{"0": 0.9831991(
25795	0	0	0	0	0.51	0	1	0	0.5867270009325953	{"0": 0.5867270(
33647	0	0	0	0	0.51	0	1	0	0.5867270009325953	{"0": 0.5867270(
20480	0	0.582	0.982	0	0.51	0	1	1	0.9758379438739092	{"0": 0.0241620(
20779	0.85	0.489	1	0	0.874	0.372	1	1	0.9895006384045648	{"0": 0.0104993(
20800	0.124	0.782	1	0	0.51	0.626	1	1	0.9426949057020769	{"0": 0.0573050(
20826	0.002	0.834	0.993	0	0.865	0	1	1	0.9954235614895207	{"0": 0.0045764:
21393	0	0.931	0.982	0	0.51	0	1	1	0.9581328588441685	{"0": 0.0418671(
21576	0	0.964	0.998	0	0.51	0	1	1	0.9649593310638387	{"0": 0.0350406(
21698	0	0.961	0.998	0	0.897	0	1	1	0.9945909774525706	{"0": 0.0054090:
21706	0	0.245	0.982	0	0.851	0	1	1	0.9938256117326599	{"0": 0.0061743(
21756	0.002	0	0.632	0	0.51	0	1	1	0.8626654951913645	{"0": 0.1373345(
21803	0.002	0	0	0	0.73	0.045	1	1	0.8281918389979603	{"0": 0.1718081(

图 2-25 “预测-3”组件的结果

第3章 生存预测

本章的两个数据集都来自泰坦尼克（Titanic）事件，标签列对应是否幸存，每个属性都有其实际的意义，便于理解、检验我们的分析结果及构造的分类模型。

上一章介绍了对于数值类型连续特征的数据集，如何使用多种分类模型进行预测。本章将会引入离散特征，譬如常见的字符串类型枚举值，有时整数型的数字也可以作为离散特征。在对离散特征进行分析时，可以借助新的工具。做模型选择时，也并不是所有分类模型都支持字符串类型。

第一个数据集是由 3 个枚举类型的特征和一个标签列构成的，通过实验可以帮助我们了解：对于这种类型的特征，如何分析、训练模型，如何进行预测。第二个数据集的构成会更复杂，同时包含数值类型与枚举类型的特征，特征的个数也更多，除了能帮助我们进一步提高使用分类模型的技巧，还会在实验中了解到机器学习领域常说的“过拟合”现象。

3.1 数据集一

数据来源于 http://www.cs.toronto.edu/~delve/data/titanic/desc.html，包括乘客和船员共 2201 人的记录，每条记录包含 4 个字段的内容，如表 3-1 所示。这个数据的特点是，所有属性列的类型都是 string 类型，都是枚举值。

表 3-1 字段含义

列 名	含 义	包含枚举值
class	阶层	1st：一等舱乘客 2nd：二等舱乘客 3rd：三等舱乘客 crew：船员
age	年纪	adult：成人 child：儿童
sex	性别	male：男 female：女
survived	是否幸存	yes：是 no：否

数据表名称为 titanic_2201，数据显示的截图如图 3-1 所示，每条记录代表一名乘客或船员，只是没有姓名等详细内容。

class	age	sex	survived
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes
1st	adult	male	yes

图 3-1 数据表内容

3.1.1 特征分析

此数据的特征都是离散特征，可以选择“离散值特征分析”组件，我们最关心“是否幸存”（survived）属性，将 survived 列作为标签列，其他 3 列（class，age，sex）看作特征属性，参数设置如图 3-2 所示。

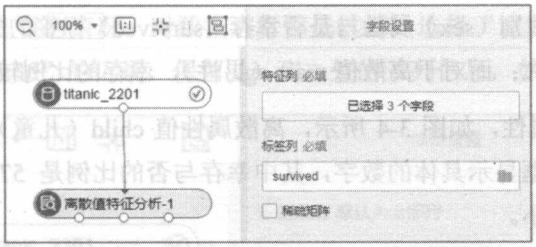


图 3-2　“离散值特征分析”组件连接及字段设置

分析结果如图 3-3 所示，分为 3 个展示区域：

1）左侧为所有计算的离散值分析图的缩略图。

2）右侧上方为某个属性列的详细分析图，每个离散值对应一个“柱体”，而每个“柱体”又根据其中样本属于不同标签（yes 和 no）的情况，标以不同颜色，左边的坐标轴对应的是样本个数；另外对于每个离散值，还计算出了基尼值和熵指标，具体数值可以参照右边的坐标轴。可以点击图例，选择显示的内容，默认是全部显示。

3）右侧下方显示的是所选属性列的整体指标，包括 Entropy、Gini、Gini Gain、Info Gain 和 Info Gain Ratio。

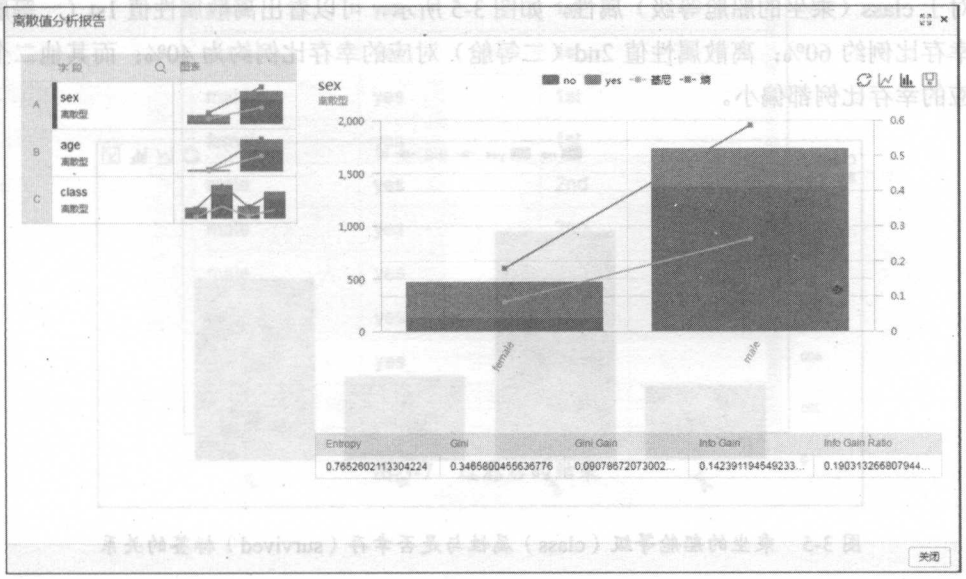


图 3-3　性别（sex）属性与是否幸存（survived）标签的关系

由图 3-3 可以看出性别（sex）属性与是否幸存（survived）标签的关系。对于离散值 female（女性），幸存的比例较大；而对于离散值 male（男性），幸存的比例较小。

对于 age（年龄）属性，如图 3-4 所示，离散属性值 child（儿童）的样本数较少，鼠标放在该区域，会出现提示框显示具体的数字，其中幸存与否的比例是 57：52；对于属性值 adult（成人），幸存的比例较小。

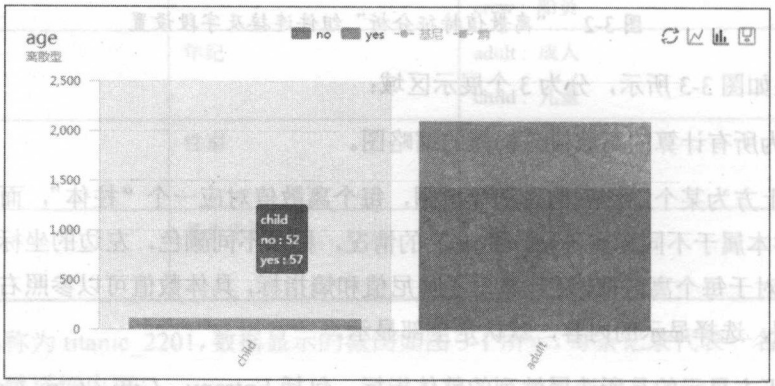


图 3-4　年龄（age）属性与是否幸存（survived）标签的关系

对于 class（乘坐的船舱等级）属性，如图 3-5 所示，可以看出离散属性值 1st（一等舱）对应的幸存比例约 60%；离散属性值 2nd（二等舱）对应的幸存比例约为 40%；而其他二个离散值对应的幸存比例都偏小。

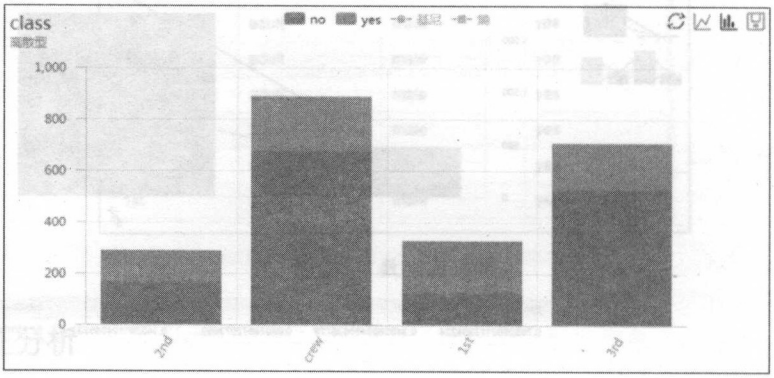


图 3-5　乘坐的船舱等级（class）属性与是否幸存（survived）标签的关系

特别地，我们关注儿童的情况，对数据进行筛选过滤，选择“过滤与映射”组件，过滤条件为 age 字段的值为“child”的记录，保留 age 列之外的三列信息，如图 3-6 所示。

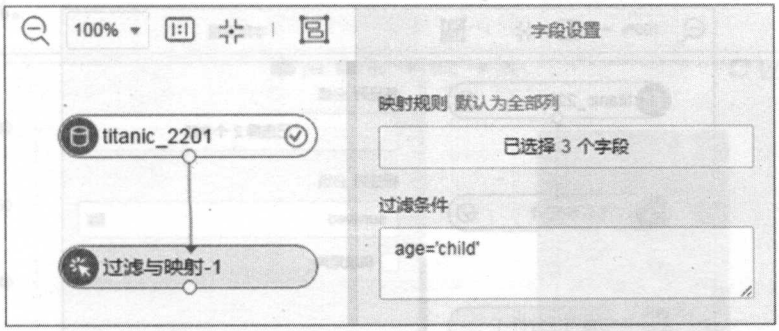


图 3-6 “过滤与映射”组件连接及字段设置

过滤后的结果如图 3-7 所示，包含 3 个字段：sex、survived 和 class。

sex ▲	survived ▲	class ▲
male	yes	1st
male	yes	1st
male	yes	1st
male	yes	1st
male	yes	1st
female	yes	1st
male	yes	2nd
male	yes	2nd
male	yes	2nd
male	yes	2nd
male	yes	2nd
male	yes	2nd

图 3-7 过滤后的结果

我们再对此数据的离散特征进行分析，如图 3-8 所示，选择特征列 sex 和 class，并选择标签列为 survived。

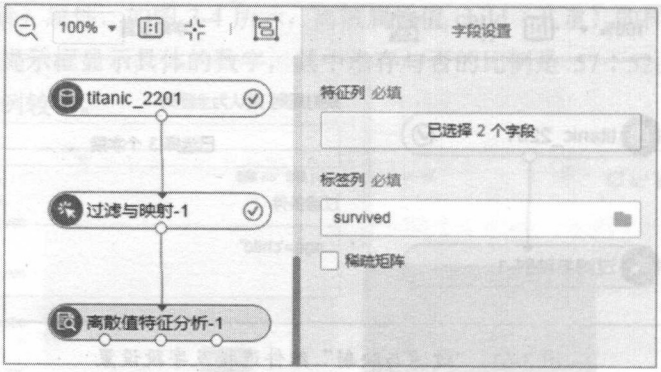


图 3-8　离散值特征分析组件连接及字段设置

性别（sex）属性的分析报告如图 3-9 所示，女孩的总数少于男孩总数；幸存下来的男孩与女孩数量差不多；女孩的幸存比例大于 50%；男孩的幸存比例小于 50%。

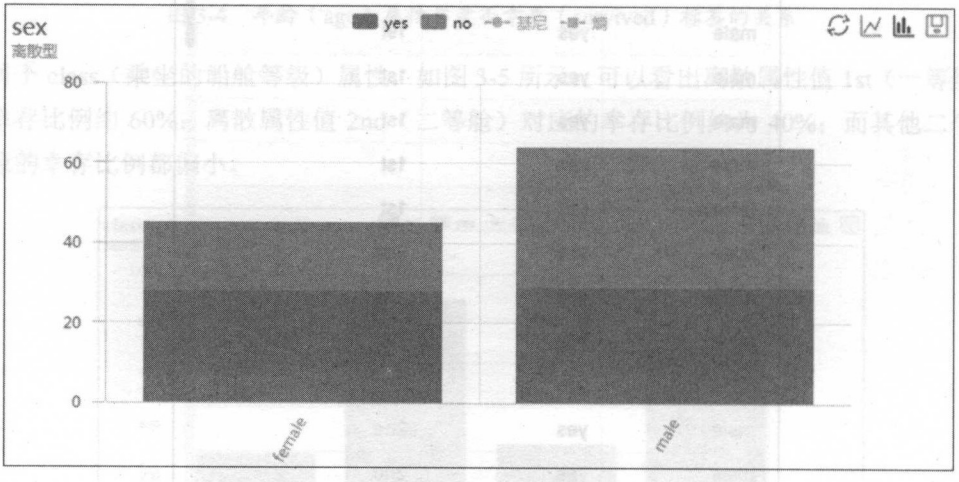


图 3-9　性别（sex）属性的分析报告

乘坐的船舱等级（class）属性的分析报告如图 3-10 所示，可以看出离散属性值 1st（一等舱）和 2nd（二等舱）对应的幸存比例都达到了 100%；离散值 3rd（三等舱）对应的幸存比例也超过了 30%。

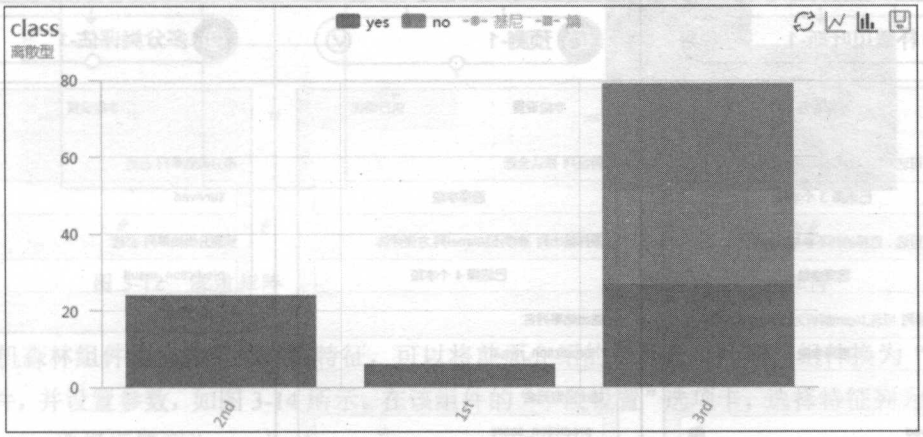


图 3-10 乘坐的船舱等级（class）属性的分析报告

3.1.2 生存预测

由前面的分析，我们会有一个直观印象，有的条件在是否生存上起到重要作用，而朴素贝叶斯算法，即假设各特征独立的情况下，通过各特征的概率乘积作为总体的概率。具体建模、预测和评估的流程如图 3-11 所示。



图 3-11 朴素贝叶斯模型流程

其中，“拆分”组件还是取 0.8 的切分比例，并设置随机数种子为 12345，便于复现此数据拆分结果。其他组件的参数设置情况如表 3-2 所示。

表 3-2　各组件参数设置

朴素贝叶斯-1	预测-1	多分类评估-1
字段设置 特征列 可选，默认除label外的全部 已选择 3 个字段 排除列 可选，选择的列不参与训练 选择字段 强制转换列 可选,bigint解析为categorical 选择字段 标签列 survived	字段设置　执行调优 特征列 默认全选 选择字段 原样输出列 推荐添加label列,方便评估 已选择 4 个字段 输出结果列名 prediction_result 输出分数列名 prediction_score 输出详细列名 prediction_detail 稀疏矩阵 格式: k1:v1, k2:v2	字段设置 原分类结果列 必选 survived 预测分类结果列 必选 prediction_result 高级选项
设置特征列为 class、age 和 sex；选择标签列为 survived	只需设置原样输出列，便于后面查看模型效果，这里选择了全部原始数据列	只要指定原分类结果列，即选择 survived 列

对测试集预测评估的混淆矩阵如图 3-12 所示，比例矩阵如图 3-13 所示，精确度为 78%，可以看到对于真实标签为 no 的测试样本，预测对的比例为 93.7%，而对于真实标签为 yes 的测试样本，预测对的比例只有 49.7%。

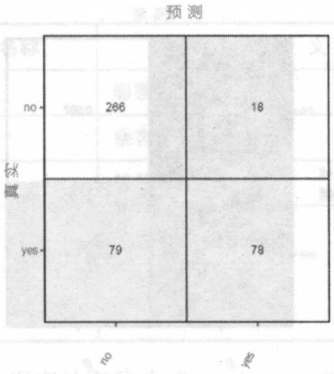


图 3-12 混淆矩阵

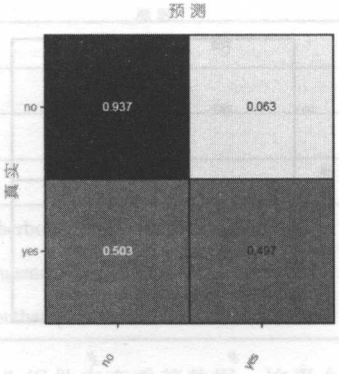


图 3-13 比例矩阵

随机森林组件也直接支持离散特征，可以将前面介绍的“朴素贝叶斯”组件换为“随机森林”组件，并设置参数，如图 3-14 所示。在该组件的“字段设置”选项卡，选择特征列为 class、age 和 sex；选择标签列为 survived。

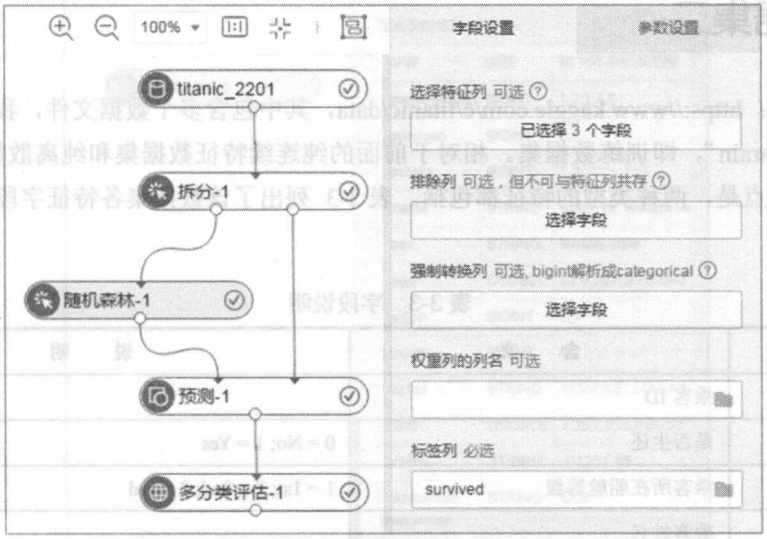


图 3-14 随机森林模型组件连接及参数设置

对测试集预测评估的混淆矩阵如图 3-15 所示，比例矩阵如图 3-16 所示，精确度为 77.3%，可以看到对于真实标签为 no 的测试样本，预测对的比例为 99.3%，而对于真实标签为 yes 的测试样本，预测对的比例只有 37.6%。

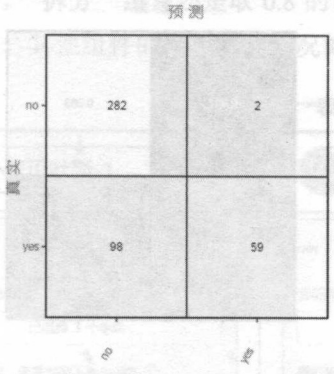


图 3-15　混淆矩阵

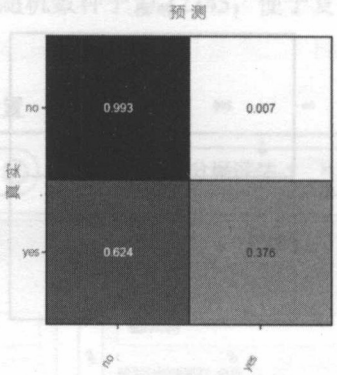


图 3-16　比例矩阵

综上，“朴素贝叶斯”组件和“随机森林”组件可以直接处理枚举类型的特征；比较简单的“朴素贝叶斯”模型在此场景下，效果不输于“随机森林”模型。

3.2　数据集二

数据来源：https://www.kaggle.com/c/titanic/data，其中包含多个数据文件，我们只需要第一个数据文件“train”，即训练数据集。相对于前面的纯连续特征数据集和纯离散特征的数据集，本数据集的特点是，两种类型的特征都包括。表 3-3 列出了该数据集各特征字段的名称、含义及说明。

表 3-3　字段说明

字段名称	含　义	说　明
passengerid	乘客 ID	
survived	是否生还	0 = No; 1 = Yes
pclass	乘客所在船舱等级	1 = 1st; 2 = 2nd; 3 = 3rd
name	乘客姓名	
sex	性别	
age	年龄	
sibsp	兄弟姐妹 / 配偶 登船人数	Number of Siblings/Spouses Aboard
parch	父母 / 孩子登船人数	Number of Parents/Children Aboard

续表

字段名称	含　义	说　明
ticket	船票号码	
fare	乘客路费	
cabin	船舱	
embarked	登船港口	C = Cherbourg 瑟堡（法国西北部港市） Q = Queenstown 昆士城 S = Southampton 南安普敦（英国港市）

该数据表的名称为“titanic_train”，通过“读数据表”组件来查看其数据。这里介绍一个小技巧：在该组件的右侧属性栏，选择“字段信息”标签，如图 3-17 所示，则可以看到每列的名称、类型、还有前 100 条记录范围内出现的不同的数据值，虽然这个信息不能代表这个数据的情况，但可以最快速地对数据有个具体的印象。



图 3-17　数据表 titanic_train 的字段信息

显示数据表的部分内容如图 3-18 所示，注意第二列 survived，标记了该乘客是否生还。

passengerid	survived	pclass	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	-	S
2	1	1	Cumings, Mrs. John Bradley (Florence ...	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O...	7.925	-	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May ...	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05	-	S
6	0	3	Moran, Mr. James	male	-	0	0	330877	8.4583	-	Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075	-	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhe...	female	27	0	2	347742	11.1333	-	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708	-	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05	-	S
14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275	-	S
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542	-	S
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16	-	S
17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125	-	Q
18	1	2	Williams, Mr. Charles Eugene	male	-	0	0	244373	13	-	S

图 3-18　数据表 titanic_train 内容显示

我们使用“全表统计”组件，进一步了解数据，如图 3-19 所示，添加“全表统计”组件，并连接数据表“titanic_train”，该组件默认会对所有数据列进行统计。

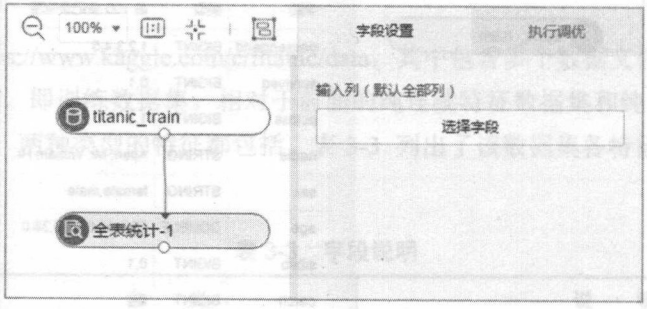


图 3-19　“全表统计”组件的连接及参数设置

统计结果如图 3-20 所示，可以看到有标签的数据记录共 891 条，在 age 字段有 177 个缺失值（missing value）。有些分类模型（譬如：朴素贝叶斯模型）不能直接处理缺失值，对于有缺失值的数据字段需要做缺失值填充处理，在后面的例子中会详细介绍。

colname	datatype	totalcount	count	missingcount	nancount	positiveinfinitycount	negativeinfinitycount	min	max	mean	variance	standarddeviation
age	double	891	714	177	0	0	0	0.42	80	29.69...	211.01912...	14.526497332334405
cabin	string	891	891	0	0	0	0	-	-	-	-	-
embarked	string	891	891	0	0	0	0	-	-	-	-	-
fare	double	891	891	0	0	0	0	0	512....	32.20...	2469.4368...	49.693428597180992
name	string	891	891	0	0	0	0	-	-	-	-	-
parch	bigint	891	891	0	0	0	0	0	6	0.381...	0.6497282...	0.8060572211299483
passenge...	bigint	891	891	0	0	0	0	1	891	446	66231	257.3538420152301
pclass	bigint	891	891	0	0	0	0	1	3	2.308...	0.6990151...	0.8360712409770492
sex	string	891	891	0	0	0	0	-	-	-	-	-
sibsp	bigint	891	891	0	0	0	0	0	8	0.523...	1.2160430...	1.102743432293432
survived	bigint	891	891	0	0	0	0	0	1	0.383...	0.2367722...	0.4865924542648575
ticket	string	891	891	0	0	0	0	-	-	-	-	-

图 3-20 数据表 titanic_train 的全表统计结果

3.2.1 随机森林模型

本节演示使用随机森林模型建模，其完整的流程如图 3-21 所示。其特点在预测评估部分，不止对测试数据进行，还对训练数据进行预测、评估。如果在训练集上预测的效果，与测试集上的预测结果有显著的差异，则我们建立的模型可能有问题。

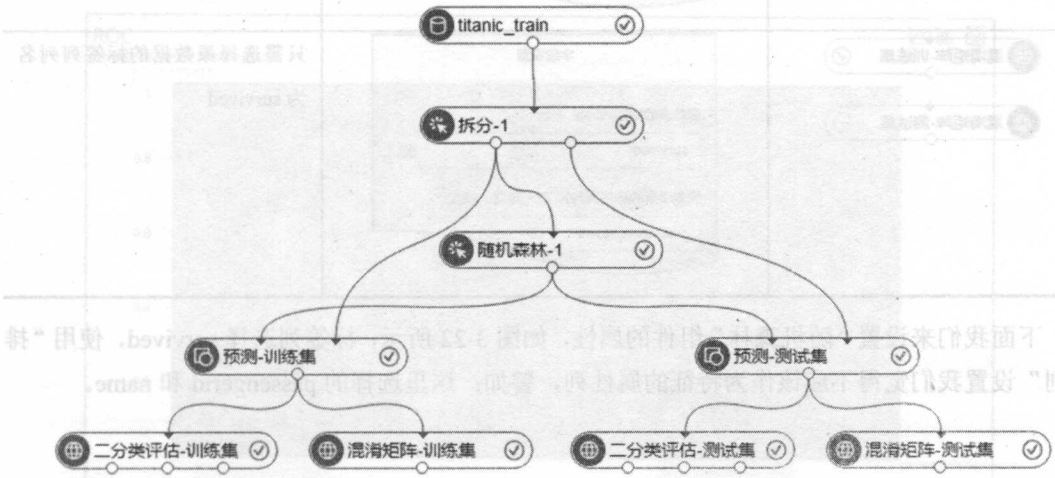


图 3-21 随机森林模型训练、预测、评估流程

其中，“拆分”组件还是取 0.8 的切分比例，并设置随机数种子为 12345，便于复现此数据拆分结果。关于预测、评估组件的设置如表 3-4 所示。

表 3-4　各组件参数设置

组　件	参数设置	说　明
预测-训练集 预测-测试集	字段设置　执行调优 特征列 默认全选 选择字段 原样输出列 推荐添加label列,方便评估 已选择 1 个字段 输出结果列名 prediction_result	只需选择原样输出列为 survived 字段，供后面的评估模块使用
二分类评估-训练集 二分类评估-测试集	字段设置 原始标签列列名 survived 分数列列名 prediction_score	只需选择原始标签列列名为 survived
混淆矩阵-训练集 混淆矩阵-测试集	字段设置 原数据的标签列列名 必选 survived 预测结果的标签列列名 若无阈值，必选 prediction_result	只需选择原数据的标签列列名为 survived

下面我们来设置“随机森林”组件的属性，如图 3-22 所示，标签列选择 survived，使用“排除列”设置我们觉得不应该作为特征的属性列，譬如：这里选择的 passengerid 和 name。

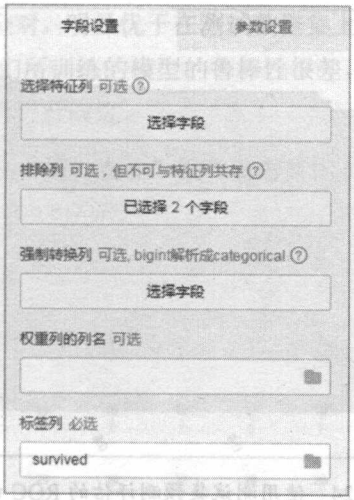


图 3-22　“随机森林”组件的字段设置

运行后，对比查看评估结果。训练集的 AUC 值为 0.999987，ROC 曲线如图 3-23 所示，整条曲线很靠近左边和上边的边线，非常理想。

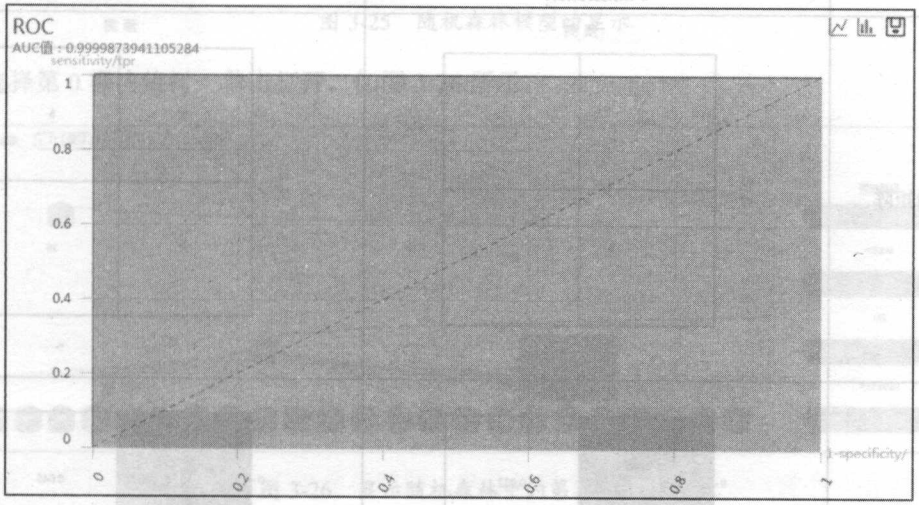


图 3-23　使用训练集预测评估的 ROC 曲线图

而对应的测试集，AUC 值 0.889，ROC 曲线如图 3-24 所示，曲线离“理想”的左上点还是有段距离。与训练集的 ROC 曲线差异明显。

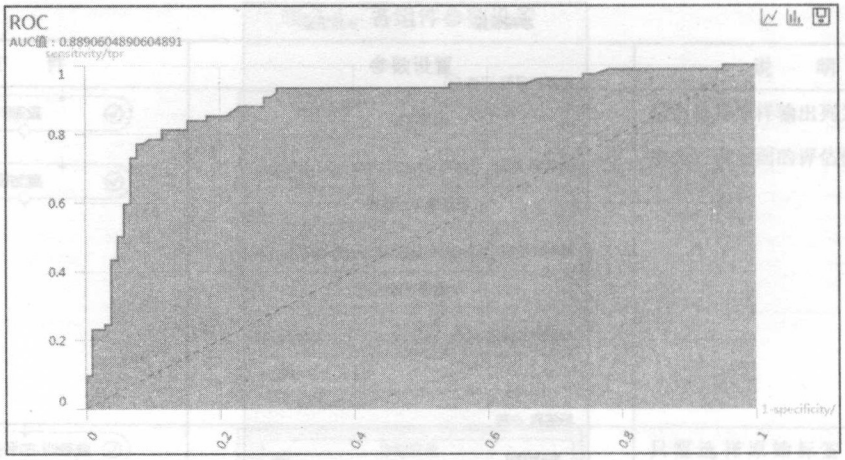


图 3-24　使用测试集预测评估的 ROC 曲线图

我们再对比训练集和测试集的混淆矩阵和比例矩阵，如表 3-5 所示。

表 3-5　训练、测试数据集的混淆矩阵和比例矩阵对比

	训练数据集	测试数据集
混淆矩阵	预测 0 1 443 1 1 267 0 1	预测 0 1 100 5 38 36 0 1
比例矩阵	预测 0 1 0.998 0.002 0.004 0.996 0 1	预测 0 1 0.952 0.048 0.514 0.486 0 1

在训练数据集的预测几乎全对，明显优于在测试数据集上的表现，这就是“过拟合”现象。“过拟合”现象的出现，说明我们所训练的模型的鲁棒性很差，应该想办法进行改善。下面我们会先分析产生此现象的原因，再进行改进。

我们查看模型寻找原因，在“随机森林”组件单击鼠标右键，选择菜单“模型选项→查看模型”，显示所有的决策树如图 3-25 所示。

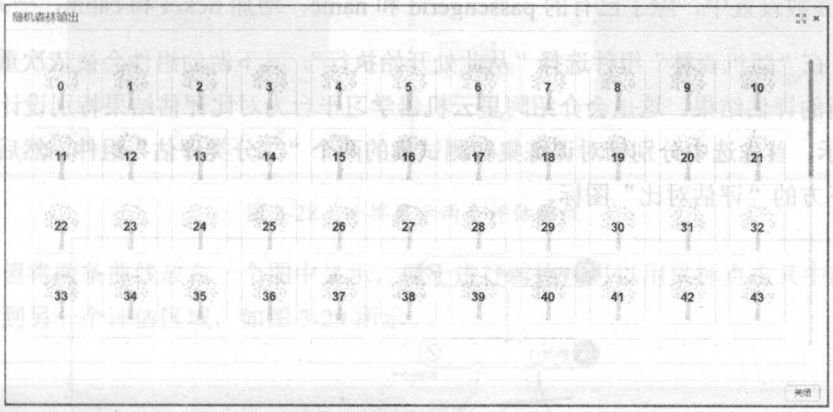


图 3-25　随机森林模型的显示

选择第 0 棵决策树，单击打开，如图 3-26 所示。

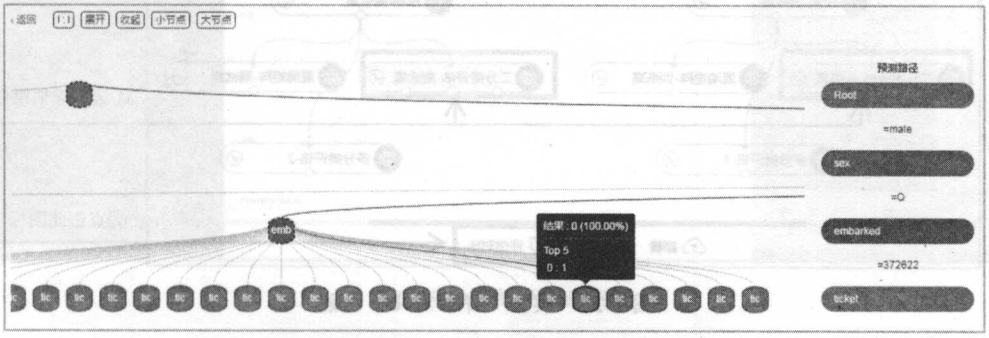


图 3-26　显示随机森林中的第 0 棵决策树

可以看到最下层有一个节点带很多子节点的情况，选取其中的一个子节点，会弹出该子节点的提示窗体，知道该子节点在训练时，只有一个标签为 0 的样本落在该子节点；查看右侧的预测路径为：[sex=male] → [embarked=Q] → [ticket=372622]。在这里我们可以猜到“过拟合”的原因了，ticket（船票号码）属性的针对性太强了，没有泛化能力，即在训练时，将训练

数据按 ticket 值进行标记，可以保证训练集样本的分类很准确，但是对测试集进行预测时，这些训练集的 ticket 值不会再出现，对预测集不会有任何帮助。这就导致了前面我们所看到的现象，对训练集的预测近乎完美，但是在测试集上反差显著。

所以我们在选择模型特征属性时，要把 ticket 属性去掉，再进一步观察其他属性，cabin 属性也属于这种情况，需要去掉。在原有的训练流程中，我们只需改动“随机森林”组件的属性，“排除列”参数设置中，除了已有的 passengerid 和 name，增加 ticket 和 cabin，一共 4 个字段。

随后，在“随机森林”组件选择“从此处开始执行”，其下游的组件会被依次重新计算。我们再查看新的评估结果。这里会介绍阿里云机器学习平台为对比评估结果特别设计的功能，如图 3-27 所示，首先选中分别针对训练集和测试集的两个“二分类评估”组件，然后，点击流程显示区域下方的“评估对比”图标。

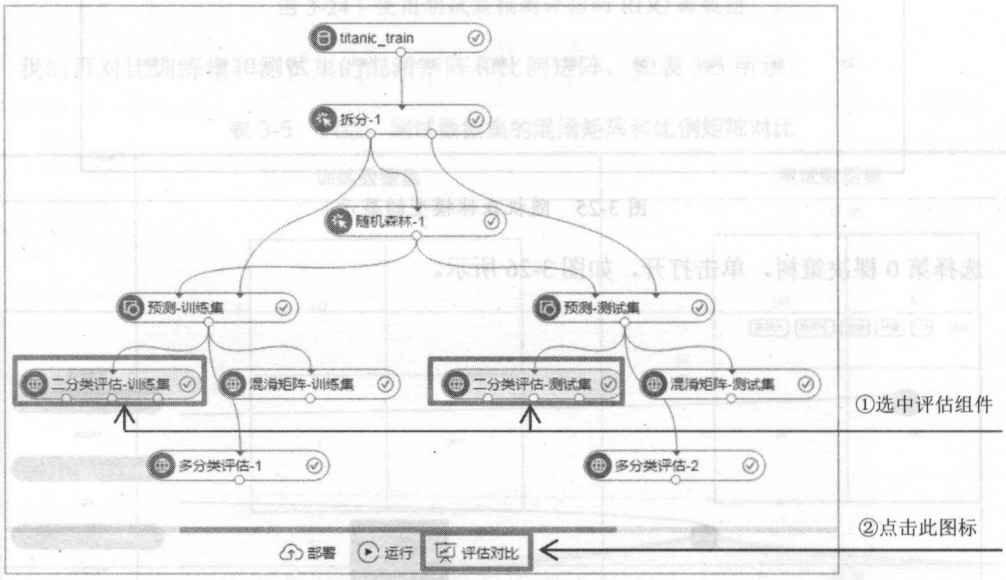


图 3-27 模型“评估对比”功能

会新弹出一个窗口，并排显示两个评估曲线，如图 3-28 所示。

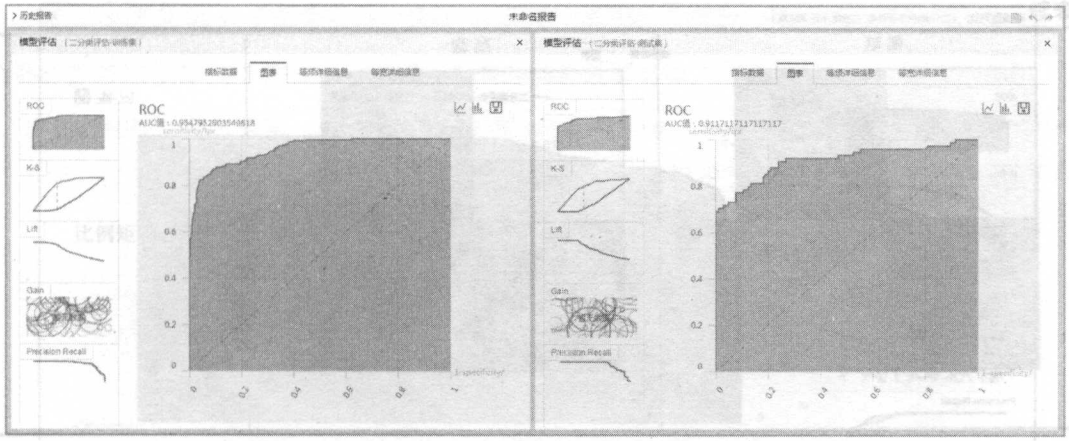


图 3-28　并排显示两个评估曲线

如果希望将两条曲线放在一个图中显示，便于进行对比，可以用鼠标点击其中的一个评估区域，拖曳到另一个评估区域，如图 3-29 所示。

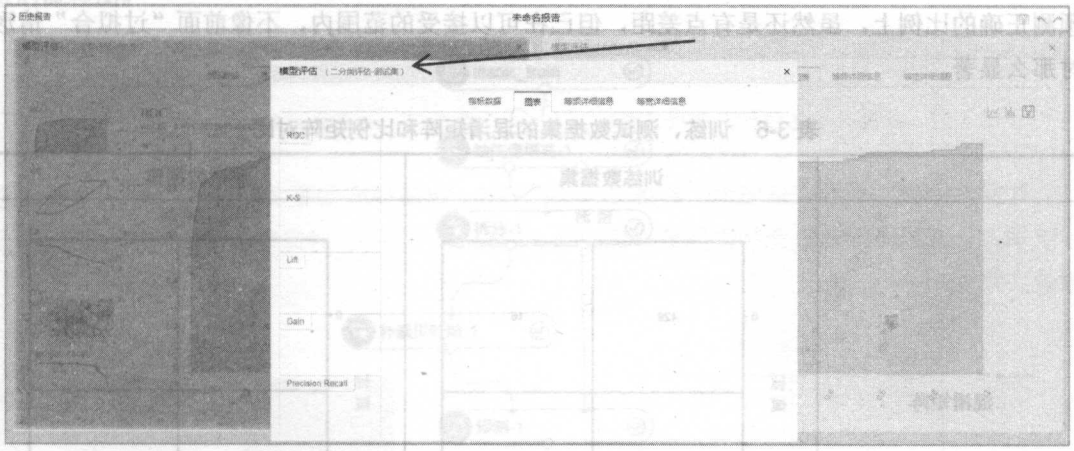


图 3-29　拖曳评估曲线

随后会重新绘制，将两个对比曲线在同一个区域显示，如图 3-30 所示。可以看到，这两个 ROC 曲线还是比较相似的，说明对于新的模型，训练集和预测集的评估结果没有明显差异，没有“过拟合”现象。

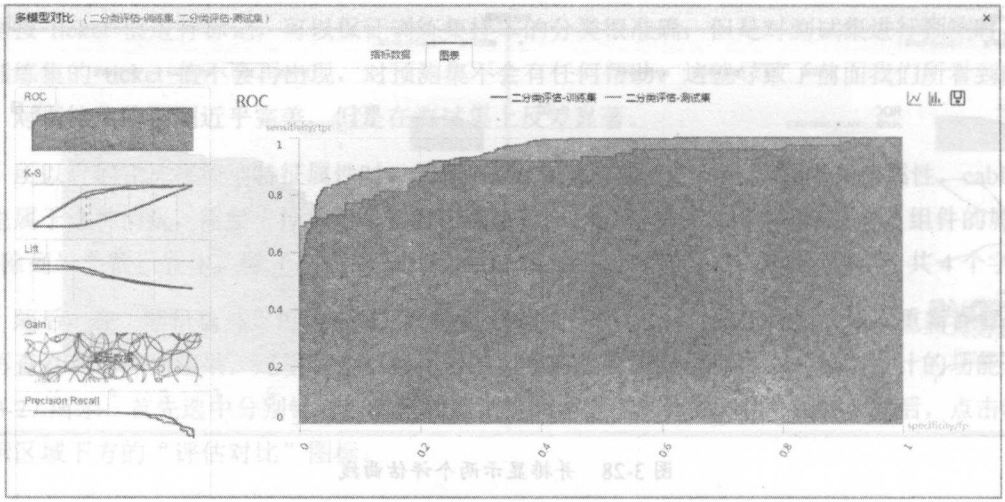


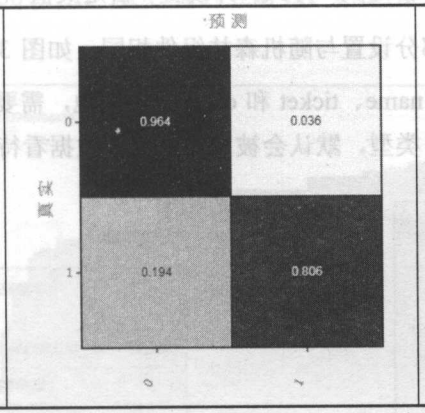
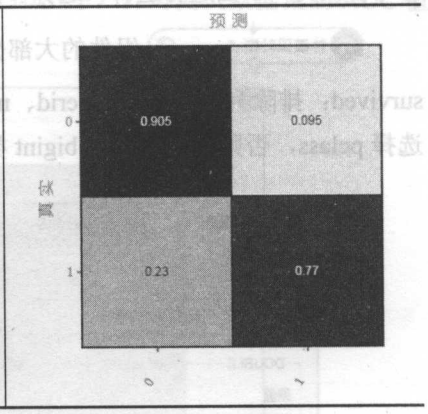
图 3-30　两个对比曲线在同一个区域显示

我们还可以通过混淆矩阵，从具体的数值上进行比较。如表 3-6 所示，训练集和测试集在预测正确的比例上，虽然还是有点差距，但已在可以接受的范围内，不像前面“过拟合”情形时那么显著。

表 3-6　训练、测试数据集的混淆矩阵和比例矩阵对比

	训练数据集	测试数据集
混淆矩阵	预测 0 1 428 16 52 216	预测 0 1 95 10 17 57

续表

比例矩阵		

3.2.2 朴素贝叶斯模型

朴素贝叶斯模型，也能同时处理连续型和离散型的属性，图 3-31 显示了使用朴素贝叶斯模型的流程图。

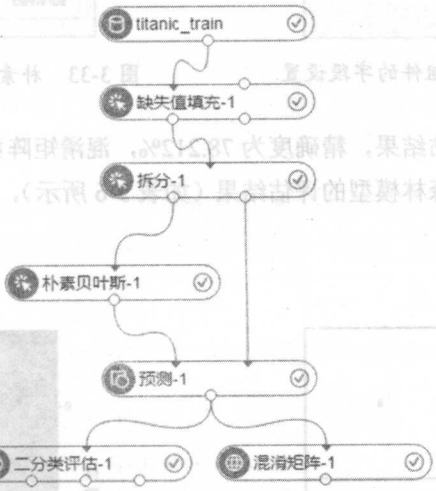
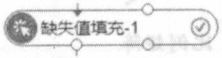


图 3-31 朴素贝叶斯模型流程

其中有两个组件需要详细介绍一下，一个是朴素贝叶斯组件；另一个是缺失值填充组件，因为原数据中含有缺失值，而朴素贝叶斯组件不能处理缺失值的情况。



组件选择含有缺失值的 age 字段，并在下面设置对缺失值的填充值，即

在“替换为”下面进行选择，选择“自定义”方式，并设置“自定义值”为0，如图3-32所示。



组件的大部分设置与随机森林组件相同，如图3-33所示，标签列选择survived；排除列选择passengerid、name、ticket和cabin。特别地，需要设置“强制转换列”，选择pclass，否则因为该列为bigint类型，默认会被当作数值型数据看待。

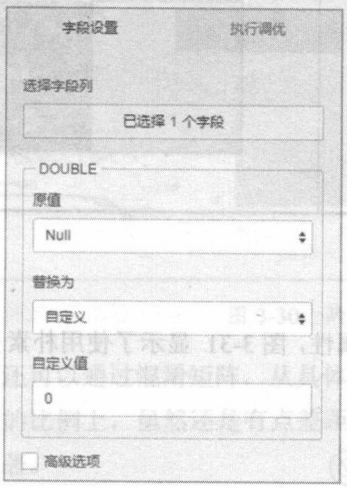


图3-32 缺失值填充组件的字段设置

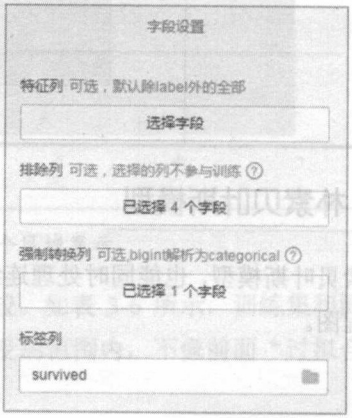


图3-33 朴素贝叶斯模型组件的字段设置

计算完成后，查看评估结果，精确度为78.212%，混淆矩阵如图3-34所示，比例矩阵如图3-35所示；其相比较随机森林模型的评估结果（如表3-6所示），对真实值为1的测试数据的预测能力很差。

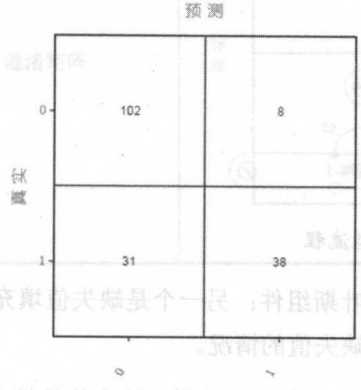


图3-34 混淆矩阵

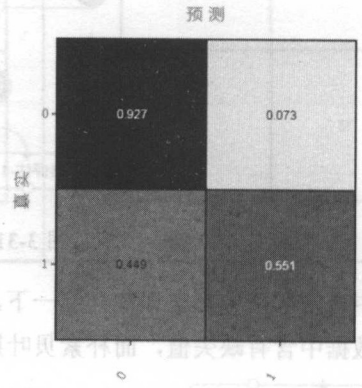


图3-35 比例矩阵

ROC 曲线如图 3-36 所示，AUC 值为 0.8158。相比图 3-28 中，针对测试数据集的 AUC 值 0.9117 还有较大差距。

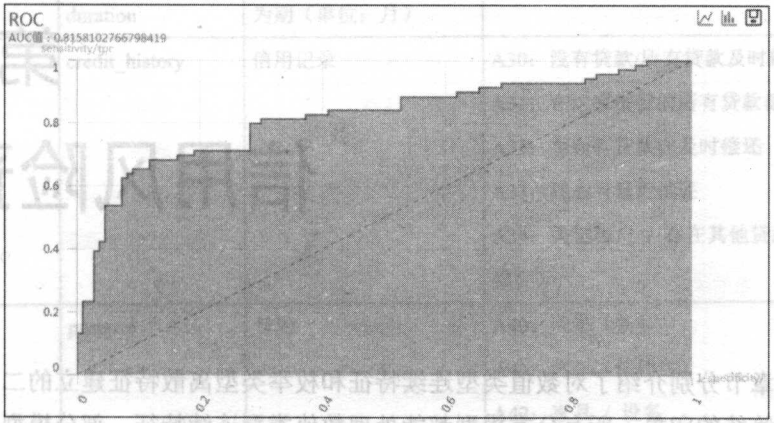


图 3-36 朴素贝叶斯模型的 ROC 曲线

第4章 信用风险预测

前面两个章节分别介绍了对数值类型连续特征和枚举类型离散特征建立的二分类模型，大家或许已有个笼统的印象，所有分类模型都能处理数值类型连续特征，部分模型能处理枚举类型离散特征。譬如：逻辑回归就是一个形式简单，效果不错的分类方法，但是只能处理数值类型数据。是否能有某种变化或方法，使逻辑回归模型也可使用枚举类型的离散特征呢？

本章将讲述特征哑元化的方法，实现特征由枚举类型向数值类型的转化，这样我们就可以使用逻辑回归等分类模型。特征哑元化方法在给我们带来惊喜的同时，也会使我们看到，特征的数量在急剧增长。面对众多的特征，哪些特征更重要些呢？是否可以减少模型特征的个数呢？本章将通过实例分析，与大家一起探讨。

本章使用的数据集为德国信用数据集（German Credit Dataset），记录了 1000 名用户的 20 个属性，并从信用风险的角度，将客户分为好客户（Good=1，按时偿还贷款）与坏客户（Bad=2，违约）两类。

下载地址：http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

数据特征属性字段说明，如表 4-1 所示。

表 4-1　字段说明

属性编号	数据类型	字段名称	含　义	备　注
1	枚举	status	现有的支票帐户的状态	A11：… < 0 马克 A12：0 <= …< 200 马克 A13：…> = 200 马克 / 工资分配为至少 1 年 A14：没有支票账户

续表

属性编号	数据类型	字段名称	含 义	备 注
2	数值	duration	为期（单位：月）	
3	枚举	credit_history	信用记录	A30：没有贷款/所有贷款及时偿还 A31：在这家银行的所有贷款都及时偿还 A32：至今有贷款在及时偿还 A33：过去有延迟偿还 A34：关键帐户 / 存在其他贷款（不是在这家银行）
4	枚举	purpose	目的	A40：汽车（新） A41：汽车（使用过） A42：家具 / 设备 A43：广播 / 电视 A44：家用电器 A45：维修 A46：教育 A47：（假期 - 不存在吗？） A48：培训 A49：商务 A410：其他
5	数值	credit_amount	信贷量	
6	枚举	savings	储蓄帐户 / 债券	A61：…< 100 马克 A62：100 <= … < 500 马克 A63：500 <= … < 1000 马克 A64：…> = 1000 马克 A65：不详 / 无人储蓄账户
7	枚举	employment	从事目前的就业	A71：失业 A72：… <1 年 A73：1 <= … <4 年 A74：4 <= … <7 年 A75：…> =7 年

续表

属性编号	数据类型	字段名称	含义	备注
8	数值	installment_rate	在可支配收入的百分比中，分期付款的比率	
9	枚举	marriage_sex	个人身份和性别	A91：男：离婚 / 分居 A92：女：离婚 / 分居 / 结婚 A93：男：单 A94：男：已婚 / 丧偶 A95：女：单
10	枚举	debtors	其他债务人/担保人	A101：无 A102：共同申请人 A103：担保人
11	数值	residence	目前住处	
12	枚举	property	财产	A121：房地产 A122：如果不是 A121：建房互助协会储蓄协定 / 人寿保险 A123：如果不是 A121 / A122：汽车或其他不在属性 6 中的 A124：未知 / 无财产
13	数值	age	年龄（单位：岁）	
14	枚举	other_plan	其他分期付款计划	A141：银行 A142：存储 A143：无
15	枚举	housing	住房	A151：租用 A152：自有 A153：免费
16	数值	number_credits	在这家银行的现有的贷款数目	

续表

属性编号	数据类型	字段名称	含　义	备　注
17	枚举	job	工作职位	A171：无业 / 不熟练 - 非居民 A172：不熟练 - 居民 A173：熟练的员工 / 公务员 A174：经理 / 个体户/高级职员/官员
18	数值	maintenance_num	应被提供赡养费的人数	
19	枚举	telephone	电话	A191：无 A192：有，以客户姓名注册
20	枚举	foreign_worker	外国工作者	A201：是 A202：无

数据表名称为“german_credit”，显示内容如图 4-1 所示，最后一列为用户分类，字段名为 class，有两个分类值：1（按时偿还贷款，好客户）与 2（违约，坏客户）。

other_plan	housing	number_credits	job	maintenance_num	telephone	foreign_worker	class
A143	A152	2	A173	1	A192	A201	1
A143	A152	1	A173	1	A191	A201	2
A143	A152	1	A172	2	A191	A201	1
A143	A153	1	A173	2	A191	A201	1
A143	A153	2	A173	2	A191	A201	2
A143	A153	1	A172	2	A192	A201	1
A143	A152	1	A173	1	A191	A201	1
A143	A151	1	A174	1	A192	A201	1

图 4-1　数据表 german_credit 的内容

4.1　整体流程

将本章所要介绍的特征哑元化及特征重要性，都整合到一个流程中，如图 4-2 所示，我们可以清晰地看到各方法在整个模型训练过程中的位置。

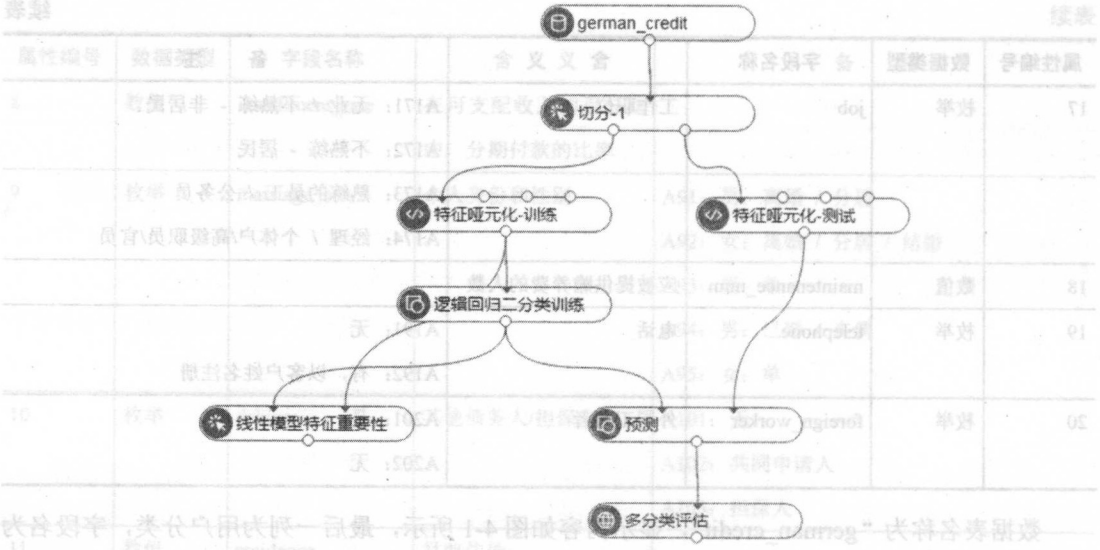


图 4-2　加入哑元化和特征重要性计算的整体流程

需要说明如下几点：

1）特征哑元化的操作是通过 SQL 组件实现的，具体方式在后面会详细介绍。

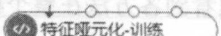
2）特征哑元化的操作，不仅要对训练数据进行，还要对所要预测的测试数据进行。

3）逻辑回归模型为线性模型，使用“线性模型特征重要性”组件，可以根据模型中各属性的系数及训练集中各特征的统计指标，得到每个属性的重要性。

下面将详细介绍算法细节。

4.1.1　特征哑元化

首先，说一下特征哑元化，是将枚举类型特征变量转换为多个二值特征变量。举个例子，如果特征属性 F 有 4 个属性值{ a,b,c,d }，则可以通过定义 4 个新的特征变量 Fa、Fb、Fc 和 Fd，使得每个新变量只能取两个值 0 或 1，表示特征属性 F 是否取到了某个属性值。如果某条记录的特征 F 的取值为 b，则特征变量 Fa=0, Fb=1,Fc=0, Fd=0。

这种变换可以通过 SQL 语句轻松实现，譬如本例中的组件 

，其对应的 SQL 为

```
select
(case status when 'A11' then 1 else 0 end) as status_A11,
(case status when 'A12' then 1 else 0 end) as status_A12,
(case status when 'A13' then 1 else 0 end) as status_A13,
(case status when 'A14' then 1 else 0 end) as status_A14,
duration,
(case credit_history when 'A30' then 1 else 0 end) as credit_history_A30,
(case credit_history when 'A31' then 1 else 0 end) as credit_history_A31,
(case credit_history when 'A32' then 1 else 0 end) as credit_history_A32,
(case credit_history when 'A33' then 1 else 0 end) as credit_history_A33,
(case credit_history when 'A34' then 1 else 0 end) as credit_history_A34,
(case purpose when 'A40' then 1 else 0 end) as purpose_A40,
(case purpose when 'A41' then 1 else 0 end) as purpose_A41,
(case purpose when 'A42' then 1 else 0 end) as purpose_A42,
(case purpose when 'A43' then 1 else 0 end) as purpose_A43,
(case purpose when 'A44' then 1 else 0 end) as purpose_A44,
(case purpose when 'A45' then 1 else 0 end) as purpose_A45,
(case purpose when 'A46' then 1 else 0 end) as purpose_A46,
(case purpose when 'A47' then 1 else 0 end) as purpose_A47,
(case purpose when 'A48' then 1 else 0 end) as purpose_A48,
(case purpose when 'A49' then 1 else 0 end) as purpose_A49,
(case purpose when 'A410' then 1 else 0 end) as purpose_A410,
credit_amount,
(case savings when 'A61' then 1 else 0 end) as savings_A61,
(case savings when 'A62' then 1 else 0 end) as savings_A62,
(case savings when 'A63' then 1 else 0 end) as savings_A63,
(case savings when 'A64' then 1 else 0 end) as savings_A64,
(case savings when 'A65' then 1 else 0 end) as savings_A65,
(case employment when 'A71' then 1 else 0 end) as employment_A71,
(case employment when 'A72' then 1 else 0 end) as employment_A72,
(case employment when 'A73' then 1 else 0 end) as employment_A73,
(case employment when 'A74' then 1 else 0 end) as employment_A74,
(case employment when 'A75' then 1 else 0 end) as employment_A75,
installment_rate,
(case marriage_sex when 'A91' then 1 else 0 end) as marriage_sex_A91,
(case marriage_sex when 'A92' then 1 else 0 end) as marriage_sex_A92,
(case marriage_sex when 'A93' then 1 else 0 end) as marriage_sex_A93,
(case marriage_sex when 'A94' then 1 else 0 end) as marriage_sex_A94,
(case marriage_sex when 'A95' then 1 else 0 end) as marriage_sex_A95,
(case debtors when 'A101' then 1 else 0 end) as debtors_A101,
(case debtors when 'A102' then 1 else 0 end) as debtors_A102,
(case debtors when 'A103' then 1 else 0 end) as debtors_A103,
residence,
(case property when 'A121' then 1 else 0 end) as property_A121,
```

```
(case property when 'A122' then 1 else 0 end) as property_A122,
(case property when 'A123' then 1 else 0 end) as property_A123,
(case property when 'A124' then 1 else 0 end) as property_A124,
age,
(case other_plan when 'A141' then 1 else 0 end) as other_plan_A141,
(case other_plan when 'A142' then 1 else 0 end) as other_plan_A142,
(case other_plan when 'A143' then 1 else 0 end) as other_plan_A143,
(case housing when 'A151' then 1 else 0 end) as housing_A151,
(case housing when 'A152' then 1 else 0 end) as housing_A152,
(case housing when 'A153' then 1 else 0 end) as housing_A153,
number_credits,
(case job when 'A171' then 1 else 0 end) as job_A171,
(case job when 'A172' then 1 else 0 end) as job_A172,
(case job when 'A173' then 1 else 0 end) as job_A173,
(case job when 'A174' then 1 else 0 end) as job_A174,
maintenance_num,
(case telephone when 'A192' then 1 else 0 end) as telephone,
(case foreign_worker when 'A201' then 1 else 0 end) as foreign_worker,
(class - 1) as class from  ${t1};
```

这里需要注意 4 个地方：

1）SQL 语句中的“case … when … then … else … end”为条件选择语句。譬如：语句“case status when 'A11' then 1 else 0 end”表示当 status 的值为'A11'时，选择 1，否则为 0。

2）对于多个属性值的 status，使用判断语句，对替换其的 4 个新属性 status_A11、status_A12，status_A13 和 status_A14 进行赋值。

3）对于只有两个属性值的 telephone 和 foreign_worker，则可以通过判断是否出现其中的一个值，用一个 0-1 二值属性替换。

4）属性 class 的取值为 1 和 2（其中，1 代表按时偿还贷款；2 表示违约），我们可以对其进行小变换，使之更符合我们的习惯，即 class 值减 1。变换后，0 代表按时偿还贷款；1 表示违约。

我们再通过对比显示，看看该操作的影响。原始的数据如图 4-3 所示；哑元化操作后的数据如图 4-4 所示。

other_plan	housing	number_credits	job	maintenance_num	telephone	foreign_worker	class
A143	A152	2	A173	1	A192	A201	1
A143	A152	1	A173	1	A191	A201	2
A143	A152	1	A172	2	A191	A201	1
A143	A153	1	A173	2	A191	A201	1
A143	A153	2	A173	2	A191	A201	2
A143	A153	1	A172	2	A192	A201	1
A143	A152	1	A173	1	A191	A201	1
A143	A151	1	A174	1	A192	A201	1

图 4-3 原始的数据

job_a171	job_a172	job_a173	job_a174	maintenance_num	telephone	foreign_worker	class
0	0	1	0	1	1	1	0
0	0	1	0	1	0	1	1
0	1	0	0	2	0	1	0
0	0	1	0	2	0	1	0
0	0	1	0	2	0	1	1
0	1	0	0	2	1	1	0
0	0	1	0	1	0	1	0
0	0	0	1	1	1	1	0

图 4-4 哑元化后的数据

从两图中横向滚动条长度的变化明显，可知哑元化后，特征的个数增加了很多。譬如，原始数据特征 job，经过哑元化操作后，产生了 4 个特征：job_a171、job_a172、job_a173、job_a174。特征 maintenance_num 是数值字段，没有受到变换的影响；特征 telephone 和 foreign_worker 的字段名称没有发生变化，但内容发生了变化，它们都是只有两个属性值的特征，通过判断是否出现其中的一个值，用一个 0-1 二值属性替换；属性 class 的原始取值为 1 和 2（其中，1 代表按时偿还贷款；2 表示违约），变换后的值为原始值减 1。变换后，0 代表按时偿还贷款，1 表示违约，更符合我们做二分类建模时对目标列的标记习惯。

4.1.2 特征重要性

模型中会涉及很多特征，但这些特征对模型的影响程度是不一样的，逻辑回归模型在机器学习中属于线性模型，我们可以通过“线性模型特征重要性”组件，得到各个特征的重要性排名。

该组件的连接比较简单，共有两个输入桩，左边的需要接入回归模型表，右边的需要接入训练模型的，为了更突出该组件的数据接入，我们对其在原流程中的位置进行了调整，如图 4-5 所示。另外，该组件只需设置一个参数，选择目标列为 class，如图 4-6 所示。

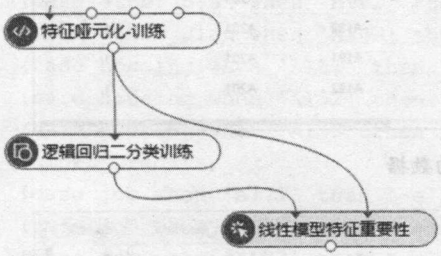


图 4-5　特征重要性组件的连接

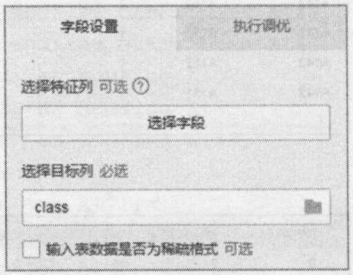


图 4-6　特征重要性组件的字段设置

待各组件运行出结果，我们先查看逻辑回归的模型，选中组件“逻辑回归二分类训练”，在弹出菜单中选择“模型选项”→“查看模型”，会弹出该模型的各个权重系数值，如图 4-7 所示；按权重值从大到小进行排列，如图 4-8 所示，最大值为 0.7639，对应的特征字段名称为 status_a11；将权重值从小到大排列，如图 4-9 所示，最小值为-1.204898，对应的特征字段名称为 status_a14。

字段名	权重	
	1	0
status_a11	0.7639456488542835	-
status_a12	0.3391255403724504	-
status_a13	-0.1210512057073507	-
status_a14	-1.204898158052929	-
duration	0.03519908841794969	-
credit_history_a30	0.3356725358453773	-
credit_history_a31	0.2738493200150684	-
credit_history_a32	-0.1053828373608862	-

图 4-7　逻辑回归模型的参数权重

字段名	权重	
	1	0
status_a11	0.7639456488542835	-
savings_a61	0.561402621664922	-
purpose_a40	0.5536619019971667	-
status_a12	0.3391255403724504	-
credit_history_a30	0.3356725358453773	-
credit_history_a31	0.2738493200150684	-
other_plan_a141	0.2691075108924015	-
purpose_a46	0.263572864055244	-

图 4-8 按权重从大到小排列

字段名	权重	
	1	0
status_a14	-1.204898158052929	-
purpose_a41	-0.62611781270894...	-
other_plan_a143	-0.61211543295269...	-
credit_history_a34	-0.61170919639366...	-
savings_a65	-0.512130436265636	-
employment_a74	-0.507021424790765	-
purpose_a43	-0.46644772303568...	-
property_a121	-0.40074596038872...	-

图 4-9 按权重从小到大排列

然后我们再选择“线性模型特征重要性”组件，在弹出菜单中选择“查看数据”，显示结果如图 4-10 所示。

colname	weight	importance
status_a11	0.7639456488542835	0.3358234059809102
status_a12	0.3391255403724504	0.1493061747211655
status_a13	0.12105120570735073	0.029860921589133407
status_a14	1.2048981580529294	0.5932535693422774
duration	0.035199088417949685	0.43087131665421263
credit_history_a30	0.3356725358453773	0.06679624949871013
credit_history_a31	0.2738493200150684	0.06042353089962566
credit_history_a32	0.10538283736088619	0.05262130365446704
credit_history_a33	0.11530799670990684	0.03217740677742094
credit_history_a34	0.6117091963936686	0.27809688000857874

图 4-10 特征重要性组件的显示结果

对比图 4-7 和图 4-10 上的数据，可以发现线性模型特征重要性中的 weight 指标就是逻辑回归模型系数的绝对值；而后面的 importance 指标，是在 weight 的基础上乘以该特征的标准差，即消除特征数值变化范围不同的影响。

通过可视化的方式，更好地显示特征重要性的差异。在“线性模型特征重要性”组件的弹出菜单中选择“查看分析报告”，可以按各特征的 weight 值进行排序，显示如图 4-11 所示，可知 weight 值较高的有：status_a14，status_a11，purpose_a41。

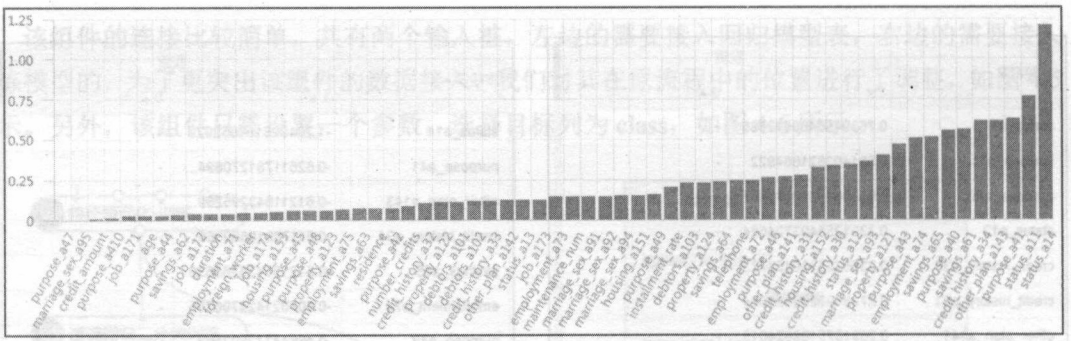


图 4-11　按各特征的 weight 值排序显示

按各特征属性的 importance 值排序，显示如图 4-12 所示，值较高的有 status_a14，duration，status_a11。

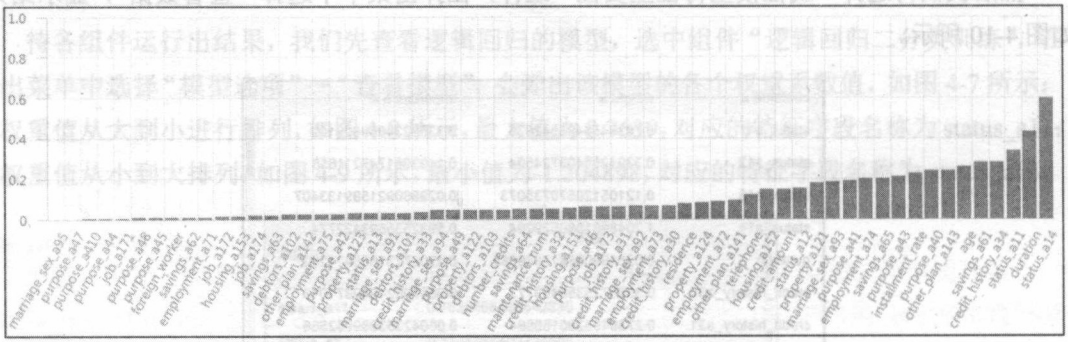


图 4-12　按各特征的 importance 值排序显示

到这里，我们可以知道哪些特征是重要的，但是特征 status_a14、status_a11 都只是标号，其对应的实际意义是什么呢？我们可以从原始数据字段说明（表 4-1），查找出经过哑元化后的特征属性字段的含义，如表 4-2 所示。

表 4-2　重要特征属性字段的说明

特征	含义
status_a14	现有的支票帐户的状态：没有支票账户
status_a11	现有的支票帐户的状态：… < 0 马克
purpose_a41	目的：汽车（使用过）
duration	为期（单位：月）

由于 duration 的标准差的值较大，使其在 weight 系数排名中的倒数第八位，一跃成为

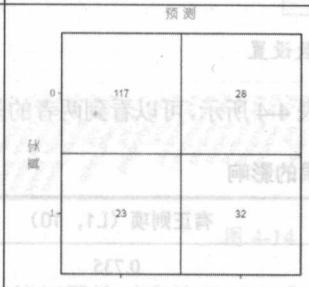
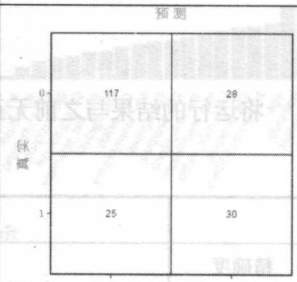
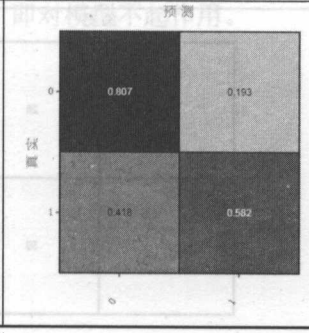
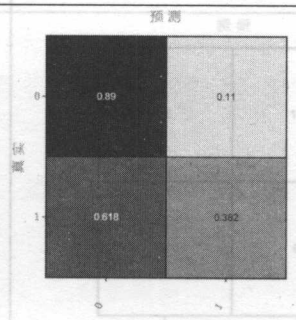
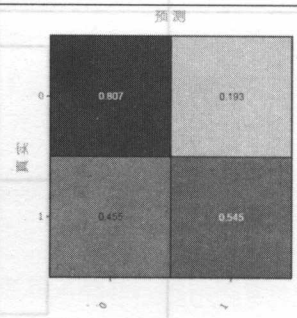
importance 系数排名中的第二名。

4.2　模型效果评估

上节给出了逻辑回归模型的处理流程图，从前面的实验可知，随机森林模型和 GBDT 模型常有更好的效果，在这个场景下，表现如何呢？

类似于前两章的做法，可以将“逻辑回归二分类”组件替换为“随机森林”或“GBDT”组件，并相应调整参数，为节约篇幅，这里不详细列出。将这三种模型的评估结果放在一起进行对比，如表 4-3 所示。观察混淆矩阵、比例矩阵，对比精确度、召回率，可以发现逻辑回归模型的总体表现最好。这一点也给我们一些启示，对特征进行哑元化，增加了特征的数量，同时哑元特征的结构简单，使得后面使用的简单分类器也可取得很好的分类效果。

表 4-3　三种模型的评估结果对比

	逻辑回归模型	随机森林模型	GBDT 模型
精确度	0.745	0.75	0.735
召回率	0.582	0.382	0.545
混淆矩阵			
比例矩阵			

4.3 减少模型特征的个数

我们在建模的时候，有时候不仅需要很高的精确度、召回率，还要考虑到模型特征的可解释性，模型的大小等因素。所以实际中还需要减少特征的个数，选取少数重要的特征建模，使得模型还具有较高的精确度、召回率等指标。

对于逻辑回归模型，可以通过使用逻辑回归模型的 L1 正则项参数，达到要求。具体的设置如图 4-13 所示，参数“正则项”选择“L1”；默认“正则系数”为 1，我们这里希望提高 L1 正则项对目标函数的影响，将此系数提升到 10；其他的参数保持默认值。

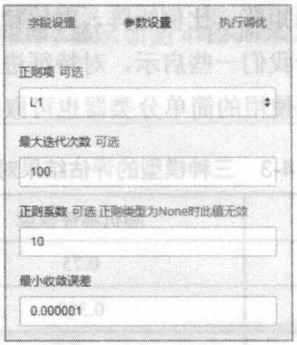
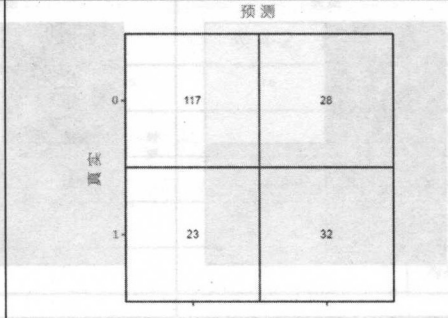
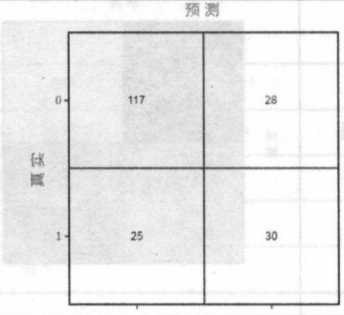


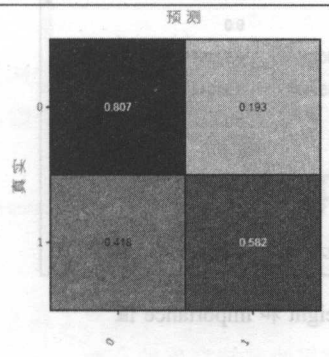
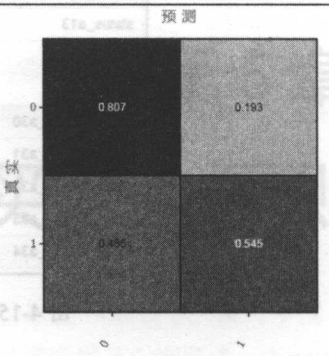
图 4-13 逻辑回归模型的参数设置

将运行的结果与之前无正则项的结果进行对比显示，如表 4-4 所示，可以看到两者的差别不大。

表 4-4 有无正则项对模型效果的影响

	无正则项	有正则项（L1，10）
精确度	0.745	0.735
召回率	0.582	0.545
混淆矩阵		

续表

	无正则项	有正则项（L1，10）
比例矩阵		

我们再通过“线性模型特征重要性”组件，查看各特征的 importance 值的排序显示，如图 4-14 所示。

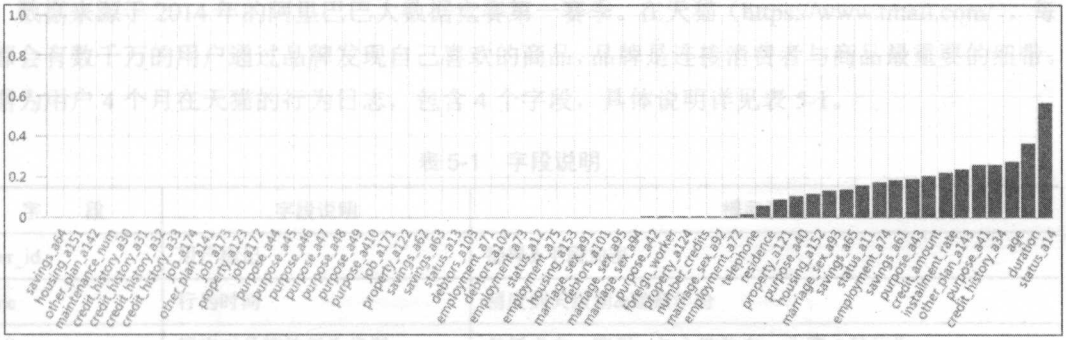


图 4-14 各特征的 importance 值的排序显示

将各特征属性对应的 weight 和 importance 值输出，如图 4-15 所示，有许多特征对应的值就是 0 值，即对模型不起作用。

colname	weight	importance
status_a11	0.3605074188823097	0.15847571024458634
status_a12	0.0	0.0
status_a13	0.0	0.0
status_a14	1.1448247299399008	0.5636753220750793
duration	0.029420939906655914	0.3601411196865224
credit_history_a30	0.0	0.0
credit_history_a31	0.0	0.0
credit_history_a32	0.0	0.0
credit_history_a33	0.0	0.0
credit_history_a34	0.5639229942380225	0.2563721882673219

图 4-15　各特征的 weight 和 importance 值

综上，使用 L1 正则的方式，明显地减少了模型所需特征的数量，便于从业务角度进行分析、解释，而且模型的分类效果没有明显下降。

第5章 用户购买行为预测

前面章节，我们一直在实验如何使用已有的特征：连续特征、离散特征、特征哑元化和选择重要特征。本章我们对“特征”的理解将更进一步，学习如何构造生成新的特征。

数据来源于2014年的阿里巴巴大数据竞赛第一赛季。在天猫（https://www.tmall.com/），每天都会有数千万的用户通过品牌发现自己喜欢的商品，品牌是连接消费者与商品最重要的纽带。数据为用户4个月在天猫的行为日志，包含4个字段，具体说明详见表5-1。

表5-1 字段说明

字　段	字段说明	提取说明
user_id	用户标记	抽样&字段加密
time	行为时间	精度到天级别&隐藏年份
action_type	用户对品牌的行为类型	包括点击、购买、加入购物车、收藏4种行为 （点击：0，购买：1，收藏：2，购物车：3）
brand_id	品牌数字ID	抽样&字段加密

用户对任意商品的行为都会映射为一行数据。其中所有商品ID都已汇总为商品对应的品牌ID。用户和品牌都分别做了一定程度的数据抽样，且数字ID都做了加密。所有行为的时间都精确到天级别（隐藏年份）。

本章将根据用户在天猫的行为日志，建立用户的品牌偏好，并预测他们将来对品牌下商品的购买行为。

我们希望预测的品牌准确率越高越好，也希望覆盖的用户和品牌越多越好，所以用最常用的准确率与召回率作为排行榜的指标。

准确率（Precision）：

$$P = \frac{\sum_{i=1}^{N} hitBrands_i}{\sum_{i=1}^{N} pBrands_i}$$

其中，N 为预测的用户数；$pBrands_i$ 为对用户 i 预测他（她）会购买的品牌列表个数；$hitBrands_i$ 为用户 i 预测的品牌列表与用户 i 真实购买的品牌交集的个数。

召回率（Recall）或称灵敏度（Sensitivity）：

$$R = \frac{\sum_{i=1}^{M} hitBrands_i}{\sum_{i=1}^{M} bBrands_i}$$

其中，M 为实际产生成交的用户数量；$bBrands_i$ 为用户 i 真实购买的品牌个数；$hitBrands_i$ 为预测的品牌列表与用户 i 真实购买的品牌交集的个数。

最后我们用 F_1-Score 来拟合准确率与召回率，并且大赛最终的比赛成绩排名以 F_1 得分为准。

$$F_1 = \frac{2 \times P \times R}{P + R}$$

5.1 数据探索

数据表名为“t_alibaba_data”，数据显示如图 5-1 所示。

user_id	brand_id	type	visit_datetime
10944750	13451	0	2014-06-04 00:00:00
10944750	13451	2	2014-06-04 00:00:00
10944750	13451	2	2014-06-04 00:00:00
10944750	13451	0	2014-06-04 00:00:00
10944750	13451	0	2014-06-04 00:00:00
10944750	13451	0	2014-06-04 00:00:00
10944750	13451	0	2014-06-04 00:00:00
10944750	13451	0	2014-06-04 00:00:00
10944750	21110	0	2014-06-07 00:00:00
10944750	1131	0	2014-07-23 00:00:00
10944750	1131	0	2014-07-23 00:00:00
10944750	8689	0	2014-05-02 00:00:00
10944750	8689	2	2014-05-02 00:00:00
10944750	8689	2	2014-05-02 00:00:00
10944750	8689	0	2014-05-02 00:00:00
10944750	8689	0	2014-05-02 00:00:00
10944750	26619	0	2014-06-28 00:00:00
10944750	5185	0	2014-07-10 00:00:00

图 5-1　数据表 t_alibaba_data 的内容

使用“全表统计”、“直方图”、“相关系数矩阵”三个组件，计算该数据集的一些基本指标。组件布局如图 5-2 所示，其中直方图组件需要选择我们关注的 type 字段，相关系数矩阵组件选择三个字段 user_id、brand_id 和 type。

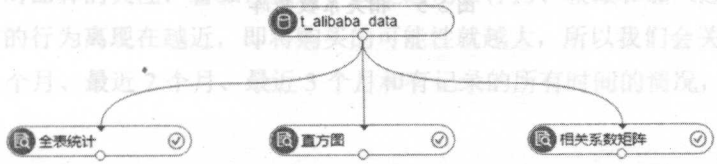


图 5-2　各统计组件布局

全表统计结果如图 5-3 所示，可以看到：

- 数据的总数为 182880 条，没有缺失值。
- 数据的开始时间为 2014.4.15，结束的时间为 2014.8.15。

colname	datatype	totalcount	count	missingcount	nancount	positiveinfinitycount	negativeinfinitycount	min	max
brand_id	bigint	182880	182880	0	0	0	0	11	29552
type	bigint	182880	182880	0	0	0	0	0	3
user_id	bigint	182880	182880	0	0	0	0	19500	12417500
visit_datetime	datetime	182880	182880	0	0	0	0	2014-04-15 00:00:00	2014-08-15 00:00:00

图 5-3　全表统计结果

type 字段共有 4 个不同的值（点击：0；购买：1；收藏：2；购物车：3），它们的直方图如图 5-4 所示。

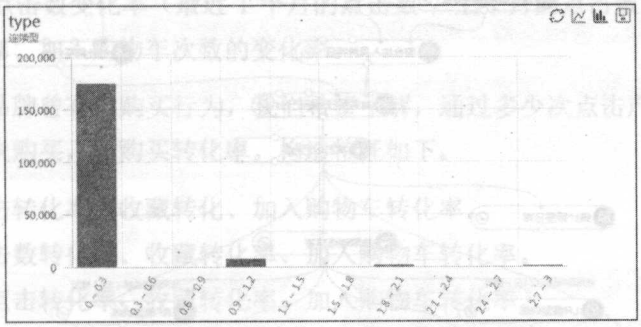


图 5-4　type 字段直方图

显然 0 值（点击行为）的个数遥遥领先。

相关系数结果如图 5-5 所示，显然各字段间不相关。

columnnames	user_id	brand_id	type
user_id	1	0.005400451879996466	0.004005357559950691
brand_id	0.005400451879996466	1	0.0035261786456922193
type	0.004005357559950691	0.0035261786456922193	1

图 5-5　相关系数矩阵

5.2　思路

预测用户在将来一个月内对品牌下商品的购买行为，即判断对于某个用户和品牌在下一个月的购买行为是否发生，也就是机器学习中典型的二分类预测问题。

解决的关键是如何得到关于用户和品牌的各种特征，并训练出二分类模型。

整体流程如图 5-6 所示，主要是构造特征和标签，使用逻辑回归（LR）和随机森林（RF）模型进行训练、预测和评估。

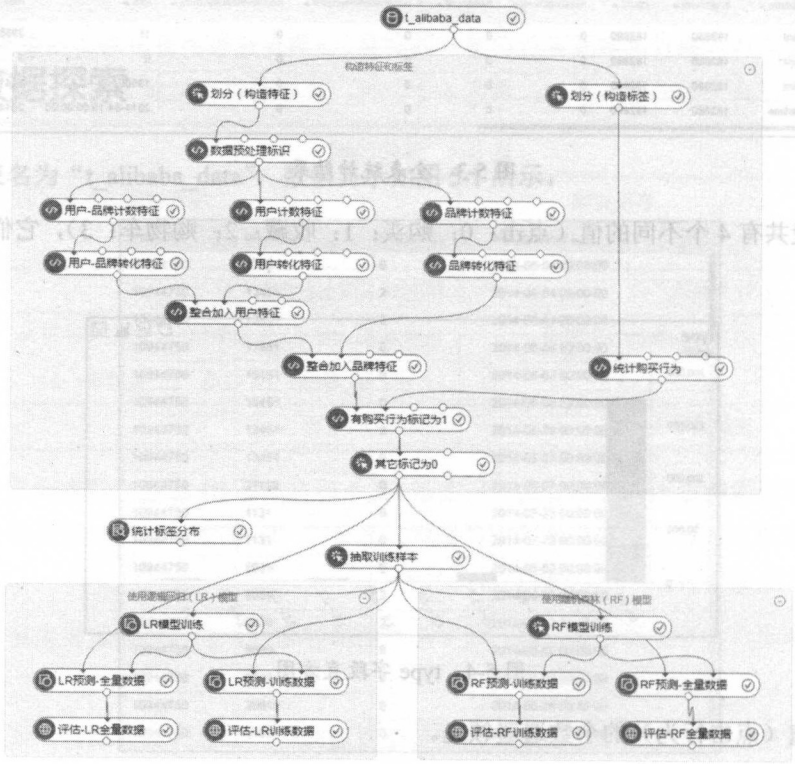


图 5-6　整体流程图

5.2.1 用户和品牌的各种特征

影响某个用户对某个品牌是否购买的特征有哪些呢？

首先是用户对品牌的关注，譬如：点击、发生过购买行为、收藏和加入过购物车。而在这些因素中，关注的行为离现在越近，即将购买的可能性就越大，所以我们会关注最近 3 天、最近 1 周、最近 1 个月、最近 2 个月、最近 3 个月和有记录的所有时间的情况，于是有了如下的一些特征。

- 最近 3 天点击数、购买数、收藏数和加入购物车次数。
- 最近 1 周点击数、购买数、收藏数和加入购物车次数。
- 最近 1 个月点击数、购买数、收藏数和加入购物车次数。
- 最近 2 个月点击数、购买数、收藏数和加入购物车次数。
- 最近 3 个月点击数、购买数、收藏数和加入购物车次数。
- 全部点击数、购买数、收藏数和加入购物车次数。

有了按时间段细分的关注次数还不够，还希望知道该数值的变化率，来刻画该关注的持续程度，我们还可以构造如下特征。

- 最近 3 天点击数变化率（最近 3 天的点击数 / 最近 4-6 天的点击数）、购买数变化率、收藏数变化率、加入购物车次数变化率。
- 最近 1 周点击数变化率（最近 1 周的点击数 / 上周的点击数）、购买数变化率、收藏数变化率、加入购物车次数的变化率。
- 最近 1 个月点击数变化率（最近 1 个月的点击数 / 上个月的点击数）、购买数变化率、收藏数变化率、加入购物车次数的变化率。

如果用户对该品牌曾有过购买行为，我们希望了解，通过多少次点击产生了一次购买，多少次收藏转化为一次购买，即购买转化率。构造特征如下。

- 最近 3 天点击转化率、收藏转化、加入购物车转化率。
- 最近 1 周点击数转化率、收藏转化率、加入购物车转化率。
- 最近 1 个月点击转化率、收藏转化率、加入购物车转化率。
- 整体的点击转化率、收藏转化率、加入购物车转化率。

其次，我们将注意力放在用户上。需要构造特征将用户的特点表现出来，重点是该用户对其关注的所有的品牌的总体行为。用户最近对所有品牌的关注度，有如下特征。

- 最近 3 天点击数、购买数、收藏数和加入购物车次数。
- 最近 1 周点击数、购买数、收藏数和加入购物车次数。
- 最近 1 个月点击数、购买数、收藏数和加入购物车次数。
- 最近 2 个月点击数、购买数、收藏数和加入购物车次数。
- 最近 3 个月点击数、购买数、收藏数和加入购物车次数。
- 全部点击数、购买数、收藏数和加入购物车次数。

每个用户都有自己的特点，有的人点击很多，却很少购买；有的人关注很久才会下单购买。这些特点可以用购买转化率来刻画。

- 最近 3 天点击转化率、收藏转化、加入购物车转化率。
- 最近 1 周点击数转化率、收藏转化率、加入购物车转化率。
- 最近 1 个月点击转化率、收藏转化率、加入购物车转化率。
- 整体的点击转化率、收藏转化率、加入购物车转化率。

最后，单独看品牌这个因素的影响。有的热门品牌，关注度很高，而我们更关心其近期的情况，有如下特征。

- 最近 3 天被点击数、被购买数、被收藏数和被加入购物车次数。
- 最近 1 周被点击数、被购买数、被收藏数和被加入购物车次数。
- 最近 1 个月被点击数、被购买数、被收藏数和被加入购物车次数。
- 最近 2 个月被点击数、被购买数、被收藏数和被加入购物车次数。
- 最近 3 个月被点击数、被购买数、被收藏数和被加入购物车次数。
- 全部被点击数、被购买数、被收藏数和被加入购物车次数。

有的品牌的受众较少，虽然没有很高的关注度，但是购买量不少，可以用转化率来描述这些特征。

- 最近 3 天点击转化率、收藏转化、加入购物车转化率。
- 最近 1 周点击数转化率、收藏转化率、加入购物车转化率。
- 最近 1 个月点击转化率、收藏转化率、加入购物车转化率。
- 整体的点击转化率、收藏转化率、加入购物车转化率。

综上，某个用户对某个品牌是否购买的特征由刻画该用户对该品牌关注的各种特征，描述该用户的特征，以及描述该品牌的特征共同构成。

5.2.2 二分类模型训练

重点是如何构造训练数据集。训练数据集中的每一条记录是关于某个用户对某个品牌的，会包含前面提到的那些特征，但同时还要有一个标签项，表示在这个特征的前提下是否会发生了购买行为。

我们知道的数据只是一些带发生时间的行为，开始时间为 2014.4.15，结束时间为 2014.8.15，一共 4 个月。怎么和特征及标签联系起来呢？

我们需要从一个新的角度来看数据，以 2014.7.15 为界将数据分为两段。由 2014.4.15 到 2014.7.15 这 3 个月的时间内发生的行为，我们可以得到用户与品牌的特征，譬如：最近 3 天点击数，最近 1 周点击转化率等；而某用户对某品牌是否会在下个月发生购买行为可以通过查询 2014.4.15 到 2014.7.15 时间段内的行为获知。

这样我们就得到了训练数据集。对于二分类模型，我们可以有很多选择，逻辑回归、随机森林等，可以根据二分类问题的特点进行选择。

5.3 计算训练数据集

由于整个构造特征、标签、模型训练和评估的流程比较长，如图 5-7 所示。首先以时间 2014-07-16 00:00:00 为界，将数据分为两份，该时间点以前的数据用来构造特征，该时间点之后的数据可以统计出在这段时间内哪些用户、品牌发生了购买行为，用来构造标签。

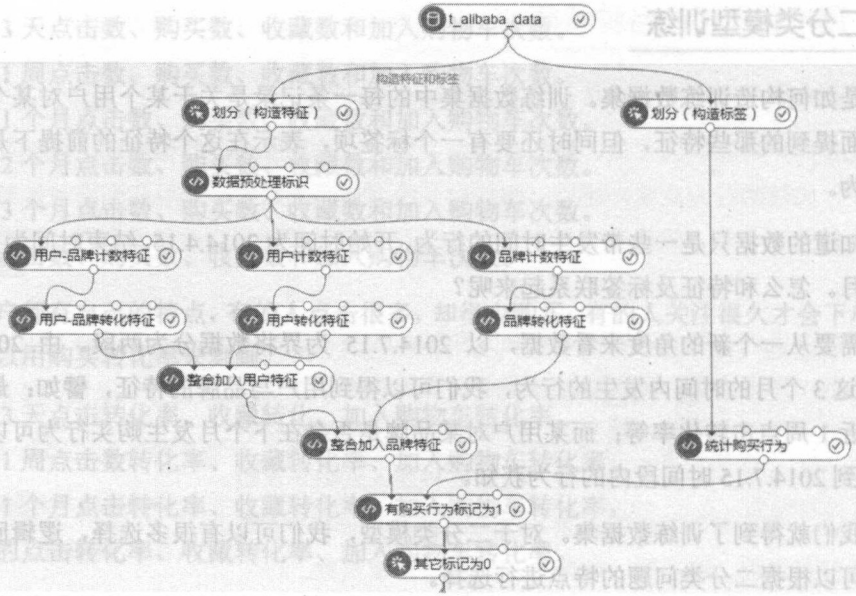


图 5-7　构造特征和标签

5.3.1　原始数据划分

要对数据按时间 2014-07-16 00:00:00 进行划分（如图 5-8 所示），需要用到条件表达式中的日期比较函数 datediff，计算两个时间 date1 和 date2 在指定时间单位 datepart 的差值。

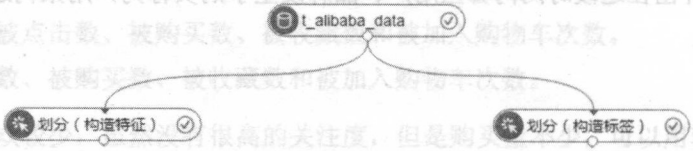


图 5-8　数据划分组件的连接

函数格式：

```
datediff(date1, date2, datepart)
```

参数说明：

datet1，date2：Datetime 类型，被减数和减数，若输入为 string 类型会隐式转换为 datetime 类型后参与运算，其他类型抛异常。

Datepart：String 类型常量。支持扩展的日期格式。若 datepart 不符合指定格式或者其他类型则会发生异常。

返回值：Bigint 类型。任一输入参数是 Null，返回 Null。

注意：计算时会按照 datepart 切掉低单位部分，然后再计算结果。

示例：

```
若start = 2005-12-31 23:59:59, end = 2006-01-01 00:00:00:
datediff(end, start, 'dd') = 1
datediff(end, start, 'mm') = 1
datediff(end, start, 'yyyy') = 1
datediff(end, start, 'hh') = 1
datediff(end, start, 'mi') = 1
datediff(end, start, 'ss') = 1
datediff(2013-05-31 13:00:00, 2013-05-31 12:30:00, 'ss') = 1800
datediff(2013-05-31 13:00:00, 2013-05-31 12:30:00, 'mi') = 30
```

利用 datediff 函数，使用“过滤与映射”组件实现数据划分，具体设置如下。

- 划分（构造特征）组件划分出 2014 年 7 月 16 日零点以前的数据，用来构造特征，如图 5-9 所示，定义过滤条件为：

```
datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>0
```

- 划分（构造标签）组件划分出 2014 年 7 月 16 日零点以后的数据，用来构造标签，如图 5-10 所示，定义过滤条件为：

```
datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=0
```



图 5-9 划分（构造特征）的字段设置

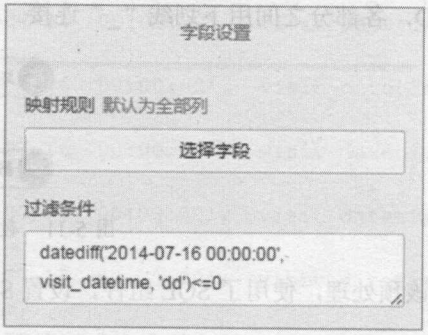


图 5-10 划分（构造标签）的字段设置

5.3.2 计算特征

我们所要计算的特征大致分为三类：某时间段内的行为发生数、转化率和变化率。后两类特征可以由相应的行为发生数相除得出，所以特征计算可以分为两个阶段：

1）计算出各时间段内的点击、购买、收藏和加入购物车的次数。

2）计算出相应的转化率和变化率。

而计算次数的时候，我们为了使计算过程清晰，并且在计算的时候有较高的效率，会通过将每个用户对品牌的行为类型和发生时间，按是否为最近 3 天发生的点击行为，是否为最近 3 天发生的购买行为……，是否为最近一个月发生的收藏行为……进行标识。如果判断为“是”，就标识为 1；否则标识为 0。进行标识后，可以将相同的用户和品牌的数据聚在同一组，将该组中所有记录对应的“是否为最近 3 天发生的点击行为”列的标识值相加，就得到了“最近 3 天发生的点击次数”。

1. 数据预处理标识

预处理环节是为后面的特征生成做准备，组件连接如图 5-11 所示，譬如：标识了近一个月内、近一周内是否有点击、购买等行为。将划分出来构造特征的数据集，生成一些新的字段，按如下规则命名：

A．以 is 开头，表明此列为标识“是否”的。

B．然后为行为信息，点击（click），购买（buy），收藏（collect）和加入购物车（cart）。

C．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，m2nd 为倒数第二个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

D．各部分之间用下划线“_”连接。

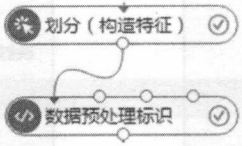


图 5-11　数据预处理标识组件连接

该预处理，使用了 SQL 组件，设置 SQL 脚本，如图 5-12 所示。

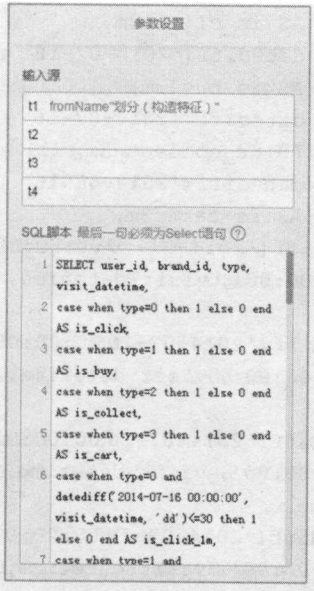


图 5-12　SQL 脚本组件的参数设置

详细的 SQL 脚本如下：

```
   SELECT user_id, brand_id, type, visit_datetime,
   case when type=0 then 1 else 0 end AS is_click,
   case when type=1 then 1 else 0 end AS is_buy,
   case when type=2 then 1 else 0 end AS is_collect,
   case when type=3 then 1 else 0 end AS is_cart,
   case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=30 then 1 else 0 end AS is_click_1m,
   case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=30 then 1 else 0 end AS is_buy_1m,
   case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=30 then 1 else 0 end AS is_collect_1m,
   case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=30 then 1 else 0 end AS is_cart_1m,
   case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=60 then 1 else 0 end AS is_click_2m,
   case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=60 then 1 else 0 end AS is_buy_2m,
   case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=60 then 1 else 0 end AS is_collect_2m,
   case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=60 then 1 else 0 end AS is_cart_2m,
   case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime,
```

```
'dd')<=90 then 1 else 0 end AS is_click_3m,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=90 then 1 else 0 end AS is_buy_3m,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=90 then 1 else 0 end AS is_collect_3m,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime,
'dd')<=90 then 1 else 0 end AS is_cart_3m,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>30
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=60 then 1 else 0 end
AS is_click_m2nd,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>30
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=60 then 1 else 0 end
AS is_buy_m2nd,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>30
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=60 then 1 else 0 end
AS is_collect_m2nd,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>30
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=60 then 1 else 0 end
AS is_cart_m2nd,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>60
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=90 then 1 else 0 end
AS is_click_m3th,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>60
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=90 then 1 else 0 end
AS is_buy_m3th,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>60
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=90 then 1 else 0 end
AS is_collect_m3th,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>60
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=90 then 1 else 0 end
AS is_cart_m3th,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=3
then 1 else 0 end AS is_click_3d,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=3
then 1 else 0 end AS is_buy_3d,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=3
then 1 else 0 end AS is_collect_3d,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=3
then 1 else 0 end AS is_cart_3d,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>3
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=6 then 1 else 0 end
AS is_click_3d2nd,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>3
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=6 then 1 else 0 end
```

```
AS is_buy_3d2nd,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>3
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=6  then 1 else 0 end
AS is_collect_3d2nd,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>3
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=6  then 1 else 0 end
AS is_cart_3d2nd,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>6
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=9  then 1 else 0 end
AS is_click_3d3th,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>6
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=9  then 1 else 0 end
AS is_buy_3d3th,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>6
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=9  then 1 else 0 end
AS is_collect_3d3th,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>6
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=9  then 1 else 0 end
AS is_cart_3d3th,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=7
then 1 else 0 end AS is_click_1w,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=7
then 1 else 0 end AS is_buy_1w,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=7
then 1 else 0 end AS is_collect_1w,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=7
then 1 else 0 end AS is_cart_1w,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>7
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=14  then 1 else 0 end
AS is_click_w2nd,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>7
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=14  then 1 else 0 end
AS is_buy_w2nd,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>7
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=14  then 1 else 0 end
AS is_collect_w2nd,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>7
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=14  then 1 else 0 end
AS is_cart_w2nd,
    case when type=0 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>14
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=21  then 1 else 0 end
AS is_click_w3th,
    case when type=1 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>14
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=21  then 1 else 0 end
```

```
AS is_buy_w3th,
    case when type=2 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>14
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=21  then 1 else 0 end
AS is_collect_w3th,
    case when type=3 and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')>14
and datediff('2014-07-16 00:00:00', visit_datetime, 'dd')<=21  then 1 else 0 end
AS is_cart_w3th
    FROM ${t1}
```

2. 用户-品牌联合特征

我们所关心的是在前三个月的行为数据中出现的用户-品牌对，而对于每个用户-品牌对，会将该用户对该品牌在前三个月的所有行为数据汇总统计、计算出如下的特征：

1）计算出各时间段内的点击、购买、收藏和加入购物车的次数。

2）计算出相应的转化率和变化率。

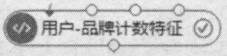
我们先介绍第 1 部分的各种“次数”。在前面计算出来的标识数据表基础上，按 user_id 和 brand_id 进行分组（GROUP），使用 SQL 中的合计函数 SUM 计算次数。得到的计数结果保存到数据表 user_brand_cnt_table_name；其中产生的列名满足如下规则：

A．以 cnt 开头，表明此列为计数信息。

B．然后为行为信息，点击 click，购买 buy，收藏 collect 和加入购物车 cart。

C．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，m2nd 为倒数第二个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

D．各部分之间用下划线“_”连接。

选择 SQL 组件 

，完成此功能。详细的 SQL 脚本如下：

```
    SELECT user_id, brand_id, SUM(is_click) as cnt_click, SUM(is_buy) as cnt_buy,
    SUM(is_collect) as cnt_collect, SUM(is_cart) as cnt_cart,
    SUM(is_click_1m) as cnt_click_1m, SUM(is_buy_1m) as cnt_buy_1m,
    SUM(is_collect_1m) as cnt_collect_1m, SUM(is_cart_1m) as cnt_cart_1m,
    SUM(is_click_2m) as cnt_click_2m, SUM(is_buy_2m) as cnt_buy_2m,
    SUM(is_collect_2m) as cnt_collect_2m, SUM(is_cart_2m) as cnt_cart_2m,
    SUM(is_click_3m) as cnt_click_3m, SUM(is_buy_3m) as cnt_buy_3m,
    SUM(is_collect_3m) as cnt_collect_3m, SUM(is_cart_3m) as cnt_cart_3m,
    SUM(is_click_m2nd) as cnt_click_m2nd, SUM(is_buy_m2nd) as cnt_buy_m2nd,
    SUM(is_collect_m2nd)    as    cnt_collect_m2nd,    SUM(is_cart_m2nd)    as
cnt_cart_m2nd,
```

```
    SUM(is_click_m3th) as cnt_click_m3th, SUM(is_buy_m3th) as cnt_buy_m3th,
    SUM(is_collect_m3th)    as    cnt_collect_m3th,    SUM(is_cart_m3th)    as
cnt_cart_m3th,
    SUM(is_click_3d) as cnt_click_3d, SUM(is_buy_3d) as cnt_buy_3d,
    SUM(is_collect_3d) as cnt_collect_3d, SUM(is_cart_3d) as cnt_cart_3d,
    SUM(is_click_3d2nd) as cnt_click_3d2nd, SUM(is_buy_3d2nd) as cnt_buy_3d2nd,
    SUM(is_collect_3d2nd)    as    cnt_collect_3d2nd,    SUM(is_cart_3d2nd)    as
cnt_cart_3d2nd,
    SUM(is_click_3d3th) as cnt_click_3d3th, SUM(is_buy_3d3th) as cnt_buy_3d3th,
    SUM(is_collect_3d3th)    as    cnt_collect_3d3th,    SUM(is_cart_3d3th)    as
cnt_cart_3d3th,
    SUM(is_click_1w) as cnt_click_1w, SUM(is_buy_1w) as cnt_buy_1w,
    SUM(is_collect_1w) as cnt_collect_1w, SUM(is_cart_1w) as cnt_cart_1w,
    SUM(is_click_w2nd) as cnt_click_w2nd, SUM(is_buy_w2nd) as cnt_buy_w2nd,
    SUM(is_collect_w2nd)    as    cnt_collect_w2nd,    SUM(is_cart_w2nd)    as
cnt_cart_w2nd,
    SUM(is_click_w3th) as cnt_click_w3th, SUM(is_buy_w3th) as cnt_buy_w3th,
    SUM(is_collect_w3th) as cnt_collect_w3th, SUM(is_cart_w3th) as cnt_cart_w3th
    FROM ${t1} GROUP BY user_id, brand_id
```

有了计数信息，可以进一步计算其转化率和变化率。我们使用 XLab 数据处理函数类中的变量转换函数 DataProc.transform(…)，将刚才计算的计数数据表 user_brand_cnt_table_name 作为输入，对其中的每一条记录进行变换，由于最终的特征中仍包括这些计算信息，所以在变换结果列中包括了原始的计数列，并将其对应的计算表达式写成该列的列名。新产生出的转化率和变化率列，满足如下规则：

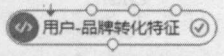
A．以 rt 开头，即 rate 的缩写，表明此列为比率信息。

B．对于变化率，中间为行为信息，点击（click），购买（buy），收藏（collect）和加入购物车（cart）。

C．对于转化率，即购买转化率，中间为行为信息加上“2buy”，点击转化率（click2buy），购买转化率（buy2buy），收藏转化率（collect2buy）和加入购物车转化率（cart2buy）。

D．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

E．各部分之间用下划线“_”连接。

选择 SQL 组件 

，完成此功能。详细的 SQL 脚本如下：

```
    SELECT user_id,brand_id,
    cnt_click,cnt_buy,cnt_collect,cnt_cart,
```

```
    cnt_click_1m,cnt_buy_1m,cnt_collect_1m,cnt_cart_1m,
    cnt_click_2m,cnt_buy_2m,cnt_collect_2m,cnt_cart_2m,
    cnt_click_3m,cnt_buy_3m,cnt_collect_3m,cnt_cart_3m,
    cnt_click_m2nd,cnt_buy_m2nd,cnt_collect_m2nd,cnt_cart_m2nd,
    cnt_click_m3th,cnt_buy_m3th,cnt_collect_m3th,cnt_cart_m3th,
    cnt_click_3d,cnt_buy_3d,cnt_collect_3d,cnt_cart_3d,
    cnt_click_3d2nd,cnt_buy_3d2nd,cnt_collect_3d2nd,cnt_cart_3d2nd,
    cnt_click_3d3th,cnt_buy_3d3th,cnt_collect_3d3th,cnt_cart_3d3th,
    cnt_click_1w,cnt_buy_1w,cnt_collect_1w,cnt_cart_1w,
    cnt_click_w2nd,cnt_buy_w2nd,cnt_collect_w2nd,cnt_cart_w2nd,
    cnt_click_w3th,cnt_buy_w3th,cnt_collect_w3th,cnt_cart_w3th,
    case when cnt_buy>cnt_click then 1.0 when cnt_buy=0 then 0.0 else
cnt_buy*1.0/cnt_click end AS rt_click2buy,
    case when cnt_buy>cnt_collect then 1.0 when cnt_buy=0 then 0.0 else
cnt_buy*1.0/cnt_collect end AS rt_collect2buy,
    case when cnt_buy>cnt_cart then 1.0 when cnt_buy=0 then 0.0 else
cnt_buy*1.0/cnt_cart end AS rt_cart2buy,
    case when cnt_buy_3d>cnt_click_3d then 1.0 when cnt_buy_3d=0 then 0.0 else
cnt_buy_3d*1.0/cnt_click_3d end AS rt_click2buy_3d,
    case when cnt_buy_3d>cnt_collect_3d then 1.0 when cnt_buy_3d=0 then 0.0 else
cnt_buy_3d*1.0/cnt_collect_3d end AS rt_collect2buy_3d,
    case when cnt_buy_3d>cnt_cart_3d then 1.0 when cnt_buy_3d=0 then 0.0 else
cnt_buy_3d*1.0/cnt_cart_3d end AS rt_cart2buy_3d,
    case when cnt_buy_1w>cnt_click_1w then 1.0 when cnt_buy_1w=0 then 0.0 else
cnt_buy_1w*1.0/cnt_click_1w end AS rt_click2buy_1w,
    case when cnt_buy_1w>cnt_collect_1w then 1.0 when cnt_buy_1w=0 then 0.0 else
cnt_buy_1w*1.0/cnt_collect_1w end AS rt_collect2buy_1w,
    case when cnt_buy_1w>cnt_cart_1w then 1.0 when cnt_buy_1w=0 then 0.0 else
cnt_buy_1w*1.0/cnt_cart_1w end AS rt_cart2buy_1w,
    case when cnt_buy_1m>cnt_click_1m then 1.0 when cnt_buy_1m=0 then 0.0 else
cnt_buy_1m*1.0/cnt_click_1m end AS rt_click2buy_1m,
    case when cnt_buy_1m>cnt_collect_1m then 1.0 when cnt_buy_1m=0 then 0.0 else
cnt_buy_1m*1.0/cnt_collect_1m end AS rt_collect2buy_1m,
    case when cnt_buy_1m>cnt_cart_1m then 1.0 when cnt_buy_1m=0 then 0.0 else
cnt_buy_1m*1.0/cnt_cart_1m end AS rt_cart2buy_1m,
    case when cnt_click_3d=0 then 0.0 when cnt_click_3d>=120.0*cnt_click_3d2nd
then 120.0 else cnt_click_3d*1.0/cnt_click_3d2nd end AS rt_click_3d,
    case when cnt_buy_3d=0 then 0.0 when cnt_buy_3d>=10.0*cnt_buy_3d2nd then 10.0
else cnt_buy_3d*1.0/cnt_buy_3d2nd end AS rt_buy_3d,
    case when cnt_collect_3d=0 then 0.0 when
cnt_collect_3d>=10.0*cnt_collect_3d2nd then 10.0 else
cnt_collect_3d*1.0/cnt_collect_3d2nd end AS rt_collect_3d,
    case when cnt_cart_3d=0 then 0.0 when cnt_cart_3d>=20.0*cnt_cart_3d2nd then
20.0 else cnt_cart_3d*1.0/cnt_cart_3d2nd end AS rt_cart_3d,
```

```
    case when cnt_click_1w=0 then 0.0 when cnt_click_1w>=300.0*cnt_click_w2nd
then 300.0 else cnt_click_1w*1.0/cnt_click_w2nd end AS rt_click_1w,
    case when cnt_buy_1w=0 then 0.0 when cnt_buy_1w>=15.0*cnt_buy_w2nd then 15.0
else cnt_buy_1w*1.0/cnt_buy_w2nd end AS rt_buy_1w,
    case        when        cnt_collect_1w=0        then        0.0        when
cnt_collect_1w>=15.0*cnt_collect_w2nd               then        15.0       else
cnt_collect_1w*1.0/cnt_collect_w2nd end AS rt_collect_1w,
    case when cnt_cart_1w=0 then 0.0 when cnt_cart_1w>=40.0*cnt_cart_w2nd then
40.0  else cnt_cart_1w*1.0/cnt_cart_w2nd end AS rt_cart_1w,
    case when cnt_click_1m=0 then 0.0 when cnt_click_1m>=500.0*cnt_click_m2nd
then 500.0 else cnt_click_1m*1.0/cnt_click_m2nd end AS rt_click_1m,
    case when cnt_buy_1m=0 then 0.0 when cnt_buy_1m>=30.0*cnt_buy_m2nd then 30.0
else cnt_buy_1m*1.0/cnt_buy_m2nd end AS rt_buy_1m,
    case        when        cnt_collect_1m=0        then        0.0        when
cnt_collect_1m>=30.0*cnt_collect_m2nd               then        30.0       else
cnt_collect_1m*1.0/cnt_collect_m2nd end AS rt_collect_1m,
    case when cnt_cart_1m=0 then 0.0 when cnt_cart_1m>=50.0*cnt_cart_m2nd then
50.0  else cnt_cart_1m*1.0/cnt_cart_m2nd end AS rt_cart_1m
    FROM ${t1}
```

3. 用户特征

这一节的重点是计算出刻画用户的特征，用户范围为前三个月的行为数据中出现的全部用户。对于每个用户，将其在前三个月的所有行为数据汇总统计、计算出如下的特征：

1）计算出各时间段内的点击、购买、收藏和加入购物车的次数。

2）计算出相应的购买转化率。

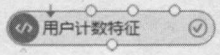
与上一节的计算方法相似，先计算第 1 部分的各种“次数”。在前面计算出来的标识数据表基础上，按 user_id 进行分组（GROUP），使用 SQL 中的合计函数 SUM 计算次数；得到的计数结果保存到数据表 user_cnt_table_name；其中产生的列名满足如下规则：

A．以 user_cnt 开头，表明此列为用户计数信息。

B．然后为行为信息，点击（click），购买（buy），收藏（collect）和加入购物车（cart）。

C．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，m2nd 为倒数第二个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

D．各部分之间用下划线“_”连接。

选择 SQL 组件 

，完成此功能。详细的 SQL 脚本如下：

```
    SELECT user_id,
    SUM(is_click) as user_cnt_click, SUM(is_buy) as user_cnt_buy,
    SUM(is_collect) as user_cnt_collect, SUM(is_cart) as user_cnt_cart,
    SUM(is_click_1m) as user_cnt_click_1m, SUM(is_buy_1m) as user_cnt_buy_1m,
    SUM(is_collect_1m)    as    user_cnt_collect_1m,    SUM(is_cart_1m)    as
user_cnt_cart_1m,
    SUM(is_click_2m) as user_cnt_click_2m, SUM(is_buy_2m) as user_cnt_buy_2m,
    SUM(is_collect_2m)    as    user_cnt_collect_2m,    SUM(is_cart_2m)    as
user_cnt_cart_2m,
    SUM(is_click_3m) as user_cnt_click_3m, SUM(is_buy_3m) as user_cnt_buy_3m,
    SUM(is_collect_3m)    as    user_cnt_collect_3m,    SUM(is_cart_3m)    as
user_cnt_cart_3m,
    SUM(is_click_m2nd)    as    user_cnt_click_m2nd,    SUM(is_buy_m2nd)    as
user_cnt_buy_m2nd,
    SUM(is_collect_m2nd)  as  user_cnt_collect_m2nd,  SUM(is_cart_m2nd)   as
user_cnt_cart_m2nd,
    SUM(is_click_m3th)    as    user_cnt_click_m3th,    SUM(is_buy_m3th)    as
user_cnt_buy_m3th,
    SUM(is_collect_m3th)  as  user_cnt_collect_m3th,  SUM(is_cart_m3th)   as
user_cnt_cart_m3th,
    SUM(is_click_3d) as user_cnt_click_3d, SUM(is_buy_3d) as user_cnt_buy_3d,
    SUM(is_collect_3d)    as    user_cnt_collect_3d,    SUM(is_cart_3d)    as
user_cnt_cart_3d,
    SUM(is_click_3d2nd)   as   user_cnt_click_3d2nd,   SUM(is_buy_3d2nd)   as
user_cnt_buy_3d2nd,
    SUM(is_collect_3d2nd) as user_cnt_collect_3d2nd, SUM(is_cart_3d2nd) as
user_cnt_cart_3d2nd,
    SUM(is_click_3d3th)   as   user_cnt_click_3d3th,   SUM(is_buy_3d3th)   as
user_cnt_buy_3d3th,
    SUM(is_collect_3d3th) as user_cnt_collect_3d3th, SUM(is_cart_3d3th) as
user_cnt_cart_3d3th,
    SUM(is_click_1w) as user_cnt_click_1w, SUM(is_buy_1w) as user_cnt_buy_1w,
    SUM(is_collect_1w)    as    user_cnt_collect_1w,    SUM(is_cart_1w)    as
user_cnt_cart_1w,
    SUM(is_click_w2nd)    as    user_cnt_click_w2nd,    SUM(is_buy_w2nd)    as
user_cnt_buy_w2nd,
    SUM(is_collect_w2nd)  as  user_cnt_collect_w2nd,  SUM(is_cart_w2nd)   as
user_cnt_cart_w2nd,
    SUM(is_click_w3th)    as    user_cnt_click_w3th,    SUM(is_buy_w3th)    as
user_cnt_buy_w3th,
    SUM(is_collect_w3th)  as  user_cnt_collect_w3th,  SUM(is_cart_w3th)   as
user_cnt_cart_w3th
    FROM ${t1} GROUP BY user_id
```

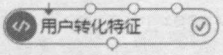
有了计数信息，再进一步计算其转化率。我们使用 XLab 数据处理函数类中的变量转换函数 DataProc.transform(...)，将刚才计算的计数数据表 user_ cnt_table_name 作为输入，对其中的每一条记录进行变换，由于最终的特征中仍包括这些计算信息，所以在变换结果列中包括了原始的计数列，并将其对应的计算表达式写成该列的列名。新产生出的购买转化率列，满足如下规则：

A．以 user_rt 开头，表明此列为用户的比率信息。

B．中间为行为信息加上“2buy”，点击转化率（click2buy），购买转化率（buy2buy），收藏转化率（collect2buy）和加入购物车转化率（cart2buy）。

C．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

D．各部分之间用下划线“_”连接。

选择 SQL 组件 

，完成此功能，详细的 SQL 脚本如下：

```
    SELECT user_id,
    user_cnt_click,user_cnt_buy,user_cnt_collect,user_cnt_cart,
    user_cnt_click_1m,user_cnt_buy_1m,user_cnt_collect_1m,user_cnt_cart_1m,
    user_cnt_click_2m,user_cnt_buy_2m,user_cnt_collect_2m,user_cnt_cart_2m,
    user_cnt_click_3m,user_cnt_buy_3m,user_cnt_collect_3m,user_cnt_cart_3m,
    user_cnt_click_m2nd,user_cnt_buy_m2nd,user_cnt_collect_m2nd,user_cnt_cart_m
2nd,
    user_cnt_click_m3th,user_cnt_buy_m3th,user_cnt_collect_m3th,user_cnt_cart_m
3th,
    user_cnt_click_3d,user_cnt_buy_3d,user_cnt_collect_3d,user_cnt_cart_3d,
    user_cnt_click_3d2nd,user_cnt_buy_3d2nd,user_cnt_collect_3d2nd,user_cnt_car
t_3d2nd,
    user_cnt_click_3d3th,user_cnt_buy_3d3th,user_cnt_collect_3d3th,user_cnt_car
t_3d3th,
    user_cnt_click_1w,user_cnt_buy_1w,user_cnt_collect_1w,user_cnt_cart_1w,
    user_cnt_click_w2nd,user_cnt_buy_w2nd,user_cnt_collect_w2nd,user_cnt_cart_w
2nd,
    user_cnt_click_w3th,user_cnt_buy_w3th,user_cnt_collect_w3th,user_cnt_cart_w
3th,
    case when user_cnt_buy>user_cnt_click then 1.0 when user_cnt_buy=0 then 0.0
else user_cnt_buy*1.0/user_cnt_click end AS user_rt_click2buy,
    case when user_cnt_buy>user_cnt_collect then 1.0 when user_cnt_buy=0 then 0.0
else user_cnt_buy*1.0/user_cnt_collect end AS user_rt_collect2buy,
    case when user_cnt_buy>user_cnt_cart then 1.0 when user_cnt_buy=0 then 0.0  else
user_cnt_buy*1.0/user_cnt_cart end AS user_rt_cart2buy,
```

```
    case when user_cnt_buy_3d>user_cnt_click_3d then 1.0 when user_cnt_buy_3d=0
then 0.0  else user_cnt_buy_3d*1.0/user_cnt_click_3d end AS user_rt_click2buy_3d,
    case when user_cnt_buy_3d>user_cnt_collect_3d then 1.0 when user_cnt_buy_3d=0
then    0.0        else    user_cnt_buy_3d*1.0/user_cnt_collect_3d    end    AS
user_rt_collect2buy_3d,
    case when user_cnt_buy_3d>user_cnt_cart_3d then 1.0 when user_cnt_buy_3d=0 then
0.0  else user_cnt_buy_3d*1.0/user_cnt_cart_3d end AS user_rt_cart2buy_3d,
    case when user_cnt_buy_1w>user_cnt_click_1w then 1.0 when user_cnt_buy_1w=0
then 0.0  else user_cnt_buy_1w*1.0/user_cnt_click_1w end AS user_rt_click2buy_1w,
    case when user_cnt_buy_1w>user_cnt_collect_1w then 1.0 when user_cnt_buy_1w=0
then    0.0        else    user_cnt_buy_1w*1.0/user_cnt_collect_1w    end    AS
user_rt_collect2buy_1w,
    case when user_cnt_buy_1w>user_cnt_cart_1w then 1.0 when user_cnt_buy_1w=0 then
0.0  else user_cnt_buy_1w*1.0/user_cnt_cart_1w end AS user_rt_cart2buy_1w,
    case when user_cnt_buy_1m>user_cnt_click_1m then 1.0 when user_cnt_buy_1m=0
then 0.0  else user_cnt_buy_1m*1.0/user_cnt_click_1m end AS user_rt_click2buy_1m,
    case when user_cnt_buy_1m>user_cnt_collect_1m then 1.0 when user_cnt_buy_1m=0
then    0.0        else    user_cnt_buy_1m*1.0/user_cnt_collect_1m    end    AS
user_rt_collect2buy_1m,
    case when user_cnt_buy_1m>user_cnt_cart_1m then 1.0 when user_cnt_buy_1m=0 then
0.0  else user_cnt_buy_1m*1.0/user_cnt_cart_1m end AS user_rt_cart2buy_1m
    FROM ${t1}
```

4. 品牌特征

这一节重点计算描述品牌的特征，品牌选择范围为前三个月的行为数据中出现的全部品牌。对于每个品牌，将前三个月中所有用户对该品牌的行为数据汇总统计、计算出如下的特征：

1）计算出各时间段内的被点击、购买、收藏和加入购物车的次数。

2）计算出相应的购买转化率。

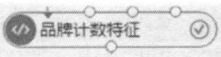
与计算用户特征的方法相似，先计算第 1 部分的各种“次数”。在前面计算出来的标识数据表基础上，按 brand_id 进行分组（GROUP），使用 SQL 中的合计函数 SUM 计算次数。得到的计数结果保存到数据表 brand_cnt_table_name；其中产生的列名满足如下规则：

A．以 brand_cnt 开头，表明此列为用户计数信息。

B．然后为行为信息，点击（click），购买（buy），收藏（collect）和加入购物车（cart）。

C．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，m2nd 为倒数第二个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

D．各部分之间用下划线“_”连接。

选择 SQL 组件 

，完成此功能，详细的 SQL 脚本如下：

```
    SELECT brand_id,
    SUM(is_click) as brand_cnt_click, SUM(is_buy) as brand_cnt_buy,
    SUM(is_collect) as brand_cnt_collect, SUM(is_cart) as brand_cnt_cart,
    SUM(is_click_1m) as brand_cnt_click_1m, SUM(is_buy_1m) as brand_cnt_buy_1m,
    SUM(is_collect_1m)    as    brand_cnt_collect_1m,    SUM(is_cart_1m)    as
brand_cnt_cart_1m,
    SUM(is_click_2m) as brand_cnt_click_2m, SUM(is_buy_2m) as brand_cnt_buy_2m,
    SUM(is_collect_2m)    as    brand_cnt_collect_2m,    SUM(is_cart_2m)    as
brand_cnt_cart_2m,
    SUM(is_click_3m) as brand_cnt_click_3m, SUM(is_buy_3m) as brand_cnt_buy_3m,
    SUM(is_collect_3m)    as    brand_cnt_collect_3m,    SUM(is_cart_3m)    as
brand_cnt_cart_3m,
    SUM(is_click_m2nd)    as    brand_cnt_click_m2nd,    SUM(is_buy_m2nd)    as
brand_cnt_buy_m2nd,
    SUM(is_collect_m2nd)  as  brand_cnt_collect_m2nd,  SUM(is_cart_m2nd)  as
brand_cnt_cart_m2nd,
    SUM(is_click_m3th)    as    brand_cnt_click_m3th,    SUM(is_buy_m3th)    as
brand_cnt_buy_m3th,
    SUM(is_collect_m3th)  as  brand_cnt_collect_m3th,  SUM(is_cart_m3th)  as
brand_cnt_cart_m3th,
    SUM(is_click_3d) as brand_cnt_click_3d, SUM(is_buy_3d) as brand_cnt_buy_3d,
    SUM(is_collect_3d)    as    brand_cnt_collect_3d,    SUM(is_cart_3d)    as
brand_cnt_cart_3d,
    SUM(is_click_3d2nd)   as   brand_cnt_click_3d2nd,   SUM(is_buy_3d2nd)   as
brand_cnt_buy_3d2nd,
    SUM(is_collect_3d2nd) as brand_cnt_collect_3d2nd, SUM(is_cart_3d2nd) as
brand_cnt_cart_3d2nd,
    SUM(is_click_3d3th)   as   brand_cnt_click_3d3th,   SUM(is_buy_3d3th)   as
brand_cnt_buy_3d3th,
    SUM(is_collect_3d3th) as brand_cnt_collect_3d3th, SUM(is_cart_3d3th) as
brand_cnt_cart_3d3th,
    SUM(is_click_1w) as brand_cnt_click_1w, SUM(is_buy_1w) as brand_cnt_buy_1w,
    SUM(is_collect_1w)    as    brand_cnt_collect_1w,    SUM(is_cart_1w)    as
brand_cnt_cart_1w,
    SUM(is_click_w2nd)    as    brand_cnt_click_w2nd,    SUM(is_buy_w2nd)    as
brand_cnt_buy_w2nd,
    SUM(is_collect_w2nd)  as  brand_cnt_collect_w2nd,  SUM(is_cart_w2nd)  as
brand_cnt_cart_w2nd,
    SUM(is_click_w3th)    as    brand_cnt_click_w3th,    SUM(is_buy_w3th)    as
brand_cnt_buy_w3th,
    SUM(is_collect_w3th)  as  brand_cnt_collect_w3th,  SUM(is_cart_w3th)  as
brand_cnt_cart_w3th
```

```
FROM ${t1} GROUP BY brand_id
```

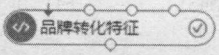
有了计数信息，再进一步计算其转化率。我们使用 XLab 数据处理函数类中的变量转换函数 DataProc.transform(...)，将刚才计算的计数数据表 brand_ cnt_table_name 作为输入，对其中的每一条记录进行变换，由于最终的特征中仍包括这些计算信息，所以在变换结果列中包括了原始的计数列，并将其对应的计算表达式写成该列的列名。新产生出的购买转化率列，满足如下规则：

A．以 brand_rt 开头，表明此列为品牌的比率信息。

B．中间为行为信息加上“2buy”，点击转化率（click2buy），购买转化率（buy2buy），收藏转化率（collect2buy）和加入购物车转化率（cart2buy）。

C．最后为时间段信息，譬如：1m 为最近 1 个月，3m 为最近 3 个月，3d 为最近 3 天，1w 为最近 1 周。若为整个时间段，则没有时间段信息。

D．各部分之间用下划线“_”连接。

选择 SQL 组件 

，完成此功能，详细的 SQL 脚本如下：

```
SELECT brand_id,
brand_cnt_click,brand_cnt_buy,brand_cnt_collect,brand_cnt_cart,
brand_cnt_click_1m,brand_cnt_buy_1m,brand_cnt_collect_1m,brand_cnt_cart_1m,
brand_cnt_click_2m,brand_cnt_buy_2m,brand_cnt_collect_2m,brand_cnt_cart_2m,
brand_cnt_click_3m,brand_cnt_buy_3m,brand_cnt_collect_3m,brand_cnt_cart_3m,
brand_cnt_click_m2nd,brand_cnt_buy_m2nd,brand_cnt_collect_m2nd,brand_cnt_ca
rt_m2nd,
brand_cnt_click_m3th,brand_cnt_buy_m3th,brand_cnt_collect_m3th,brand_cnt_ca
rt_m3th,
brand_cnt_click_3d,brand_cnt_buy_3d,brand_cnt_collect_3d,brand_cnt_cart_3d,
brand_cnt_click_3d2nd,brand_cnt_buy_3d2nd,brand_cnt_collect_3d2nd,brand_cnt
_cart_3d2nd,
brand_cnt_click_3d3th,brand_cnt_buy_3d3th,brand_cnt_collect_3d3th,brand_cnt
_cart_3d3th,
brand_cnt_click_1w,brand_cnt_buy_1w,brand_cnt_collect_1w,brand_cnt_cart_1w,
brand_cnt_click_w2nd,brand_cnt_buy_w2nd,brand_cnt_collect_w2nd,brand_cnt_ca
rt_w2nd,
brand_cnt_click_w3th,brand_cnt_buy_w3th,brand_cnt_collect_w3th,brand_cnt_ca
rt_w3th,
case when brand_cnt_buy>brand_cnt_click then 1.0 when brand_cnt_buy=0 then 0.0
else brand_cnt_buy*1.0/brand_cnt_click end AS brand_rt_click2buy,
case when brand_cnt_buy>brand_cnt_collect then 1.0 when brand_cnt_buy=0 then
0.0  else brand_cnt_buy*1.0/brand_cnt_collect end AS brand_rt_collect2buy,
```

```
    case when brand_cnt_buy>brand_cnt_cart then 1.0 when brand_cnt_buy=0 then 0.0
else brand_cnt_buy*1.0/brand_cnt_cart end AS brand_rt_cart2buy,
    case when brand_cnt_buy_3d>brand_cnt_click_3d then 1.0 when brand_cnt_buy_3d=0
then    0.0        else    brand_cnt_buy_3d*1.0/brand_cnt_click_3d    end    AS
brand_rt_click2buy_3d,
    case    when    brand_cnt_buy_3d>brand_cnt_collect_3d    then    1.0    when
brand_cnt_buy_3d=0 then 0.0  else brand_cnt_buy_3d*1.0/brand_cnt_collect_3d end
AS brand_rt_collect2buy_3d,
    case when brand_cnt_buy_3d>brand_cnt_cart_3d then 1.0 when brand_cnt_buy_3d=0
then    0.0        else    brand_cnt_buy_3d*1.0/brand_cnt_cart_3d    end    AS
brand_rt_cart2buy_3d,
    case when brand_cnt_buy_1w>brand_cnt_click_1w then 1.0 when brand_cnt_buy_1w=0
then    0.0        else    brand_cnt_buy_1w*1.0/brand_cnt_click_1w    end    AS
brand_rt_click2buy_1w,
    case    when    brand_cnt_buy_1w>brand_cnt_collect_1w    then    1.0    when
brand_cnt_buy_1w=0 then 0.0  else brand_cnt_buy_1w*1.0/brand_cnt_collect_1w end
AS brand_rt_collect2buy_1w,
    case when brand_cnt_buy_1w>brand_cnt_cart_1w then 1.0 when brand_cnt_buy_1w=0
then 0.0  else brand_cnt_buy_1w*1.0/brand_cnt_cart_1w end AS brand_cart2buy_1w,
    case when brand_cnt_buy_1m>brand_cnt_click_1m then 1.0 when brand_cnt_buy_1m=0
then    0.0        else    brand_cnt_buy_1m*1.0/brand_cnt_click_1m    end    AS
brand_rt_click2buy_1m,
    case    when    brand_cnt_buy_1m>brand_cnt_collect_1m    then    1.0    when
brand_cnt_buy_1m=0 then 0.0  else brand_cnt_buy_1m*1.0/brand_cnt_collect_1m end
AS brand_rt_collect2buy_1m,
    case when brand_cnt_buy_1m>brand_cnt_cart_1m then 1.0 when brand_cnt_buy_1m=0
then 0.0  else brand_cnt_buy_1m*1.0/brand_cnt_cart_1m end AS brand_rt_cart2buy_1m
    FROM ${t1}
```

5. 整合训练数据的特征

前面分别计算出了用户-品牌联合特征、用户特征和品牌特征，在判断一个用户是否会购买一个品牌的时候，用户对该品牌的关注程度是重要的因素，可以通过用户-品牌联合特征进行刻画；另外，该用户的购买特点和该品牌被购买的情况，也是重要的因素。所以，我们需要将所有的特征整合起来进行分析。

整合的方法如图 5-13 所示，是以 user_id 和 brand_id 作为联合主键，包括用户-品牌联合特征，再使用 SQL 的 JOIN（连接）操作，将 user_id 对应的用户特征加入，将 brand_id 对应的品牌特征加入。



图 5-13　整合训练数据的特征

其中，SQL 组件  的 SQL 脚本为：

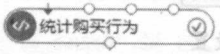

```
   SELECT  a.*  ,  b.user_cnt_click,  b.user_cnt_buy,  b.user_cnt_collect,
b.user_cnt_cart,          b.user_cnt_click_1m,          b.user_cnt_buy_1m,
b.user_cnt_collect_1m,       b.user_cnt_cart_1m,       b.user_cnt_click_2m,
b.user_cnt_buy_2m,        b.user_cnt_collect_2m,       b.user_cnt_cart_2m,
b.user_cnt_click_3m,       b.user_cnt_buy_3m,       b.user_cnt_collect_3m,
b.user_cnt_cart_3m,       b.user_cnt_click_m2nd,       b.user_cnt_buy_m2nd,
b.user_cnt_collect_m2nd,     b.user_cnt_cart_m2nd,     b.user_cnt_click_m3th,
b.user_cnt_buy_m3th,      b.user_cnt_collect_m3th,      b.user_cnt_cart_m3th,
b.user_cnt_click_3d,       b.user_cnt_buy_3d,       b.user_cnt_collect_3d,
b.user_cnt_cart_3d,       b.user_cnt_click_3d2nd,       b.user_cnt_buy_3d2nd,
b.user_cnt_collect_3d2nd,    b.user_cnt_cart_3d2nd,    b.user_cnt_click_3d3th,
b.user_cnt_buy_3d3th,     b.user_cnt_collect_3d3th,     b.user_cnt_cart_3d3th,
b.user_cnt_click_1w,       b.user_cnt_buy_1w,       b.user_cnt_collect_1w,
b.user_cnt_cart_1w,       b.user_cnt_click_w2nd,       b.user_cnt_buy_w2nd,
b.user_cnt_collect_w2nd,     b.user_cnt_cart_w2nd,     b.user_cnt_click_w3th,
b.user_cnt_buy_w3th,      b.user_cnt_collect_w3th,      b.user_cnt_cart_w3th,
b.user_rt_click2buy,       b.user_rt_collect2buy,       b.user_rt_cart2buy,
b.user_rt_click2buy_3d,    b.user_rt_collect2buy_3d,    b.user_rt_cart2buy_3d,
b.user_rt_click2buy_1w,    b.user_rt_collect2buy_1w,    b.user_rt_cart2buy_1w,
b.user_rt_click2buy_1m, b.user_rt_collect2buy_1m, b.user_rt_cart2buy_1m
   FROM ${t1} a JOIN ${t2} b ON a.user_id=b.user_id
```

SQL 组件  的 SQL 脚本为：


```
   SELECT  a.*  ,  b.brand_cnt_click,  b.brand_cnt_buy,  b.brand_cnt_collect,
b.brand_cnt_cart,          b.brand_cnt_click_1m,          b.brand_cnt_buy_1m,
b.brand_cnt_collect_1m,       b.brand_cnt_cart_1m,       b.brand_cnt_click_2m,
b.brand_cnt_buy_2m,        b.brand_cnt_collect_2m,       b.brand_cnt_cart_2m,
b.brand_cnt_click_3m,       b.brand_cnt_buy_3m,       b.brand_cnt_collect_3m,
b.brand_cnt_cart_3m,       b.brand_cnt_click_m2nd,       b.brand_cnt_buy_m2nd,
b.brand_cnt_collect_m2nd,    b.brand_cnt_cart_m2nd,    b.brand_cnt_click_m3th,
b.brand_cnt_buy_m3th,     b.brand_cnt_collect_m3th,     b.brand_cnt_cart_m3th,
b.brand_cnt_click_3d,       b.brand_cnt_buy_3d,       b.brand_cnt_collect_3d,
```

```
b.brand_cnt_cart_3d,      b.brand_cnt_click_3d2nd,      b.brand_cnt_buy_3d2nd,
b.brand_cnt_collect_3d2nd,  b.brand_cnt_cart_3d2nd,  b.brand_cnt_click_3d3th,
b.brand_cnt_buy_3d3th,    b.brand_cnt_collect_3d3th,    b.brand_cnt_cart_3d3th,
b.brand_cnt_click_1w,       b.brand_cnt_buy_1w,       b.brand_cnt_collect_1w,
b.brand_cnt_cart_1w,      b.brand_cnt_click_w2nd,       b.brand_cnt_buy_w2nd,
b.brand_cnt_collect_w2nd,    b.brand_cnt_cart_w2nd,    b.brand_cnt_click_w3th,
b.brand_cnt_buy_w3th,     b.brand_cnt_collect_w3th,     b.brand_cnt_cart_w3th,
b.brand_rt_click2buy,       b.brand_rt_collect2buy,       b.brand_rt_cart2buy,
b.brand_rt_click2buy_3d,  b.brand_rt_collect2buy_3d,  b.brand_rt_cart2buy_3d,
b.brand_rt_click2buy_1w,    b.brand_rt_collect2buy_1w,    b.brand_cart2buy_1w,
b.brand_rt_click2buy_1m, b.brand_rt_collect2buy_1m, b.brand_rt_cart2buy_1m
   FROM ${t1} a JOIN ${t2} b ON a.brand_id=b.brand_id
```

5.3.3 计算标签

首先计算出最后一个月中发生了购买行为的用户-品牌对，可以使用 SQL 语句轻松得到，使用 SQL 组件 

，其 SQL 语句为：

```
SELECT DISTINCT user_id, brand_id, 1 as label FROM ${t1} WHERE type=1
```

如果上一节计算出来的特征数据表中某一条记录的 user_id 和 brand_id 在刚才计算出来的数据表 next_month_buy_table_name 中出现，就说明该 user_id 用户在最后一个月购买了该 brand_id 品牌，可以标记为 1；反之就没有购买。流程如图 5-14 所示。

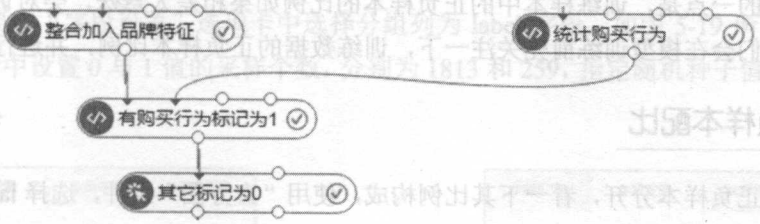
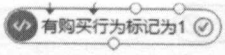
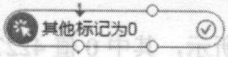


图 5-14 计算标签的流程

其中，

为 SQL 组件，使用 LEFT OUTER JOIN（左连接）方法，将有购买行为的那些记录的 label 列的赋值为 1，其他项为缺失值状态。SQL 语句如下：

```
   SELECT a.*, b.label FROM ${t1} a LEFT OUTER JOIN ${t2} b ON a.user_id=b.user_id
and a.brand_id=b.brand_id
```

然后，我们再使用缺失值填充组件 

，选择 label 字段，缺失值用 0 填充，设置如图 5-15 所示。

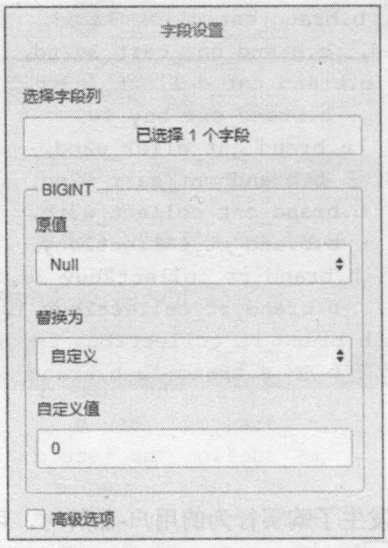


图 5-15　缺失值填充组件字段设置

5.4　二分类模型训练

前面已经构造出很多特征，本节将尝试多种二分类模型，并选出最适合的模型。这里需要提醒用户注意的一点是，训练样本中的正负样本的比例如果相差太悬殊，会对训练出来的模型产生影响。我们会在模型训练前先关注一下，训练数据的正负样本比例，并进行调整。

5.4.1　正负样本配比

我们先将正负样本分开，看一下其比例构成，使用“直方图”组件，选择 label 字段，如图 5-16 所示。

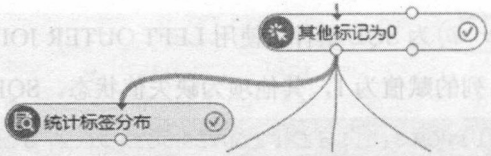


图 5-16 直方图组件的连接

计算结果直方图显示如图 5-17 所示，其中 0 值 42272 个，1 值 259 个。

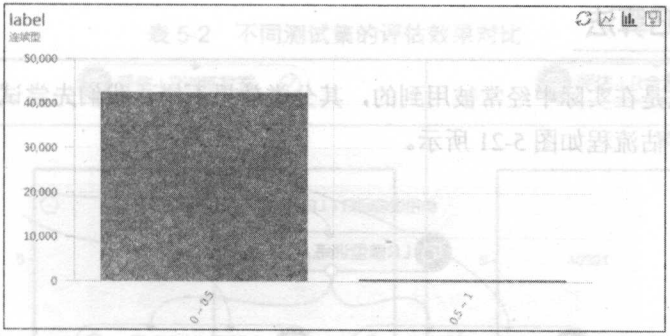


图 5-17　直方图结果

显然，正负样本的数量相差悬殊。这里我们会用随机抽样的办法，获取 7 倍正样本数量的负样本，并将其与正样本合在一起，作为后续二分类模型的训练数据。该功能使用“分层采样”组件实现，如图 5-18 所示。

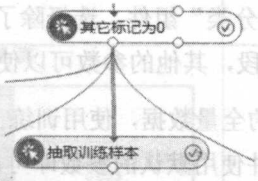


图 5-18　连接分层采样组件抽取训练样本

该组件需要在“字段设置”选项卡中选择分组列为 label 字段，如图 5-19 所示；并在“参数设置”选项卡中设置 0 与 1 值的采样个数，分别为 1813 和 259，指定随机种子值为 1234567，如图 5-20 所示。



图 5-19　组件的字段设置

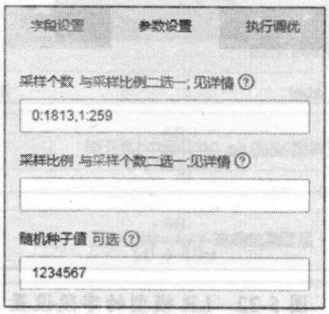


图 5-20　组件的参数设置

注：这里所用的 1 : 7 的正负样本比例不是绝对的，用户可以尝试不同的比例。

5.4.2 逻辑回归算法

逻辑回归算法是在实际中经常被用到的，其分类效果不错，我们先尝试一下该算法，具体的训练、预测和评估流程如图 5-21 所示。

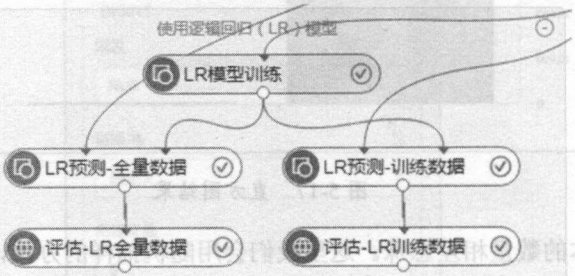


图 5-21 LR 模型的训练、预测和评估

LR 模型训练使用“逻辑回归二分类”组件，选择除了 user_id、brand_id 和 label 外的其他字段作为特征列，目标列为 label 字段，其他的参数可以使用默认参数，如图 5-22 所示。

然后分别对训练数据和采样前的全量数据，使用训练出来的逻辑回归（LR）模型使用“预测”组件进行预测，这两个预测组件使用默认的参数即可。然后分别将这两个预测结果连接到“评估”组件查看模型的效果，评估组件的参数只要设置原分类结果列为 label，如图 5-23 所示。

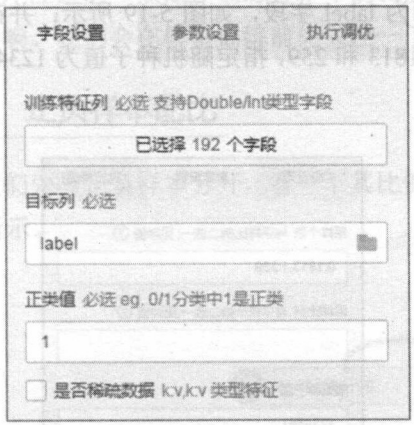


图 5-22 LR 模型的字段设置

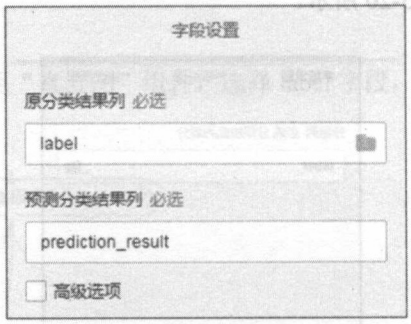
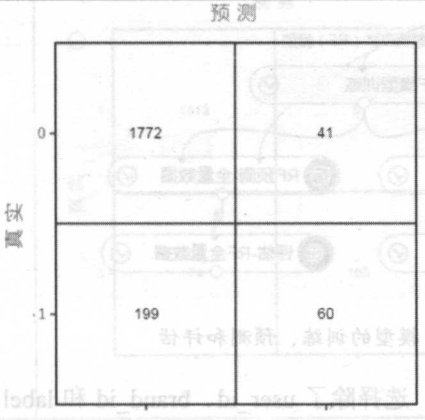
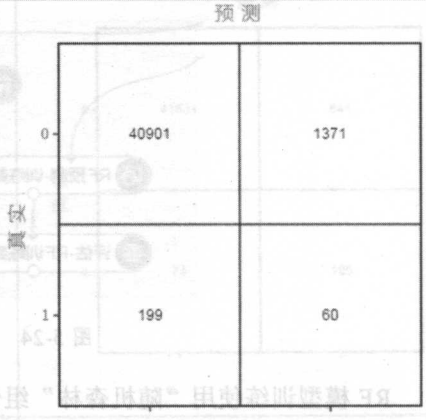
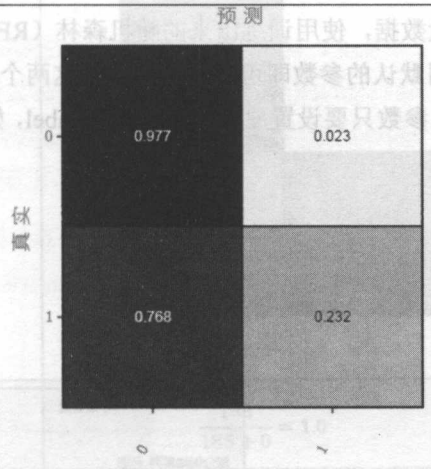
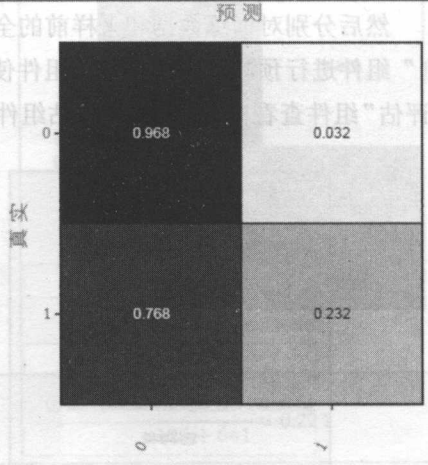


图 5-23 “评估”组件的字段设置

我们查看评估的结果，如表 5-2 所示。

表 5-2　不同测试集的评估效果对比

	评估-LR训练数据	评估-LR全量数据
混淆矩阵		
比例矩阵		
准确率（Precision）	$\frac{60}{60+41} \approx 0.594$	$\frac{60}{60+1371} \approx 0.042$
召回率（Recall）	$\frac{60}{60+199} \approx 0.232$	$\frac{60}{60+199} \approx 0.232$
F_1-Score $=\frac{2\times P\times R}{P+R}$	0.333	0.071

5.4.3 随机森林算法

我们再尝试随机森林算法，具体的训练、预测和评估流程如图 5-24 所示。

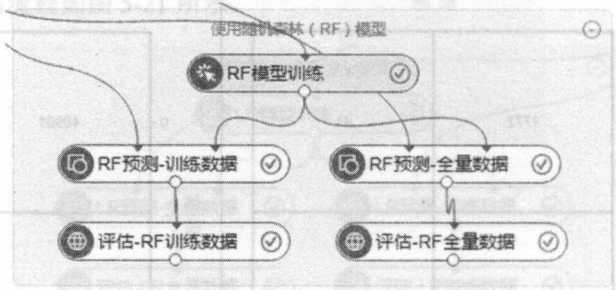


图 5-24 RF 模型的训练、预测和评估

RF 模型训练使用“随机森林”组件，选择除了 user_id、brand_id 和 label 外的其他字段作为特征列，标签列为 label 字段，其他的参数可以使用默认参数，如图 5-25 所示。

然后分别对训练数据和采样前的全量数据，使用训练出来的随机森林（RF）模型使用“预测”组件进行预测，这两个预测组件使用默认的参数即可。然后分别将这两个预测结果连接到“评估”组件查看模型的效果，评估组件的参数只要设置原分类结果列为 label，如图 5-26 所示。

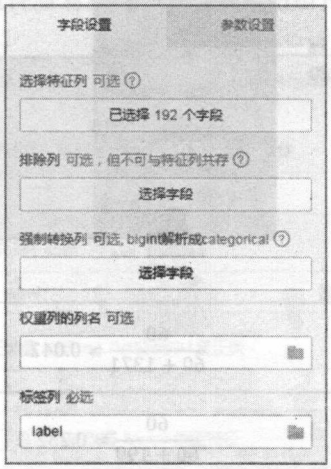


图 5-25 “随机森林”组件的字段设置

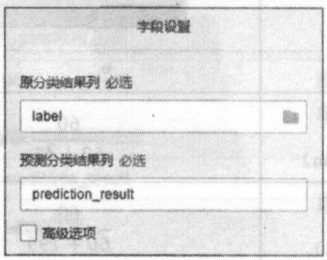
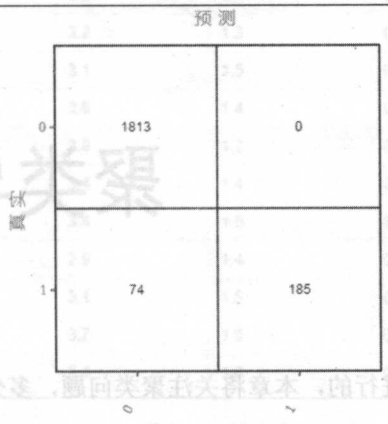
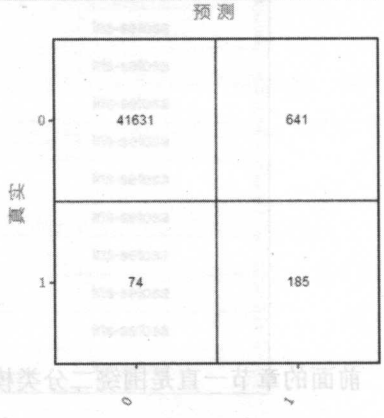
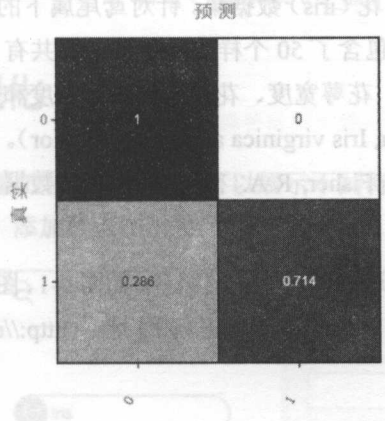
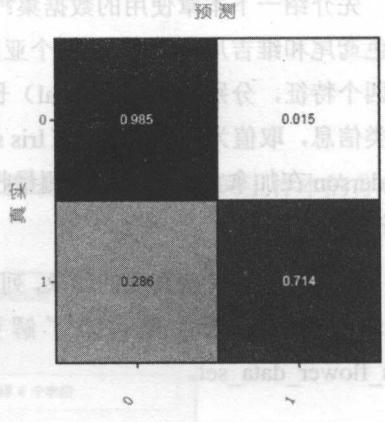


图 5-26 “评估”组件的字段设置

我们查看评估的结果，如表 5-3 所示。

表 5-3　不同测试集的评估效果对比

	评估-RF训练数据	评估-RF全量数据
混淆矩阵		
比例矩阵		
准确率（Precision）	$\frac{185}{185+0}=1.0$	$\frac{185}{185+641}\approx 0.224$
召回率（Recall）	$\frac{185}{185+74}\approx 0.714$	$\frac{185}{185+74}\approx 0.714$
$F_1\text{-Score}=\frac{2\times P\times R}{P+R}$	0.833	0.341

显然，各项指标上明显优于逻辑回归算法。我们可以将随机森林模型作为最终的预测模型。

第6章 聚类与分类

前面的章节一直是围绕二分类模型进行的，本章将关注聚类问题，多分类问题。

先介绍一下本章使用的数据集：鸢尾花（iris）数据集。针对鸢尾属下的三个亚属（山鸢尾、变色鸢尾和维吉尼亚鸢尾），每个亚属各包含了 50 个样本，该数据集共有 150 条数据。数据集有四个特征，分别是花萼（Sepal）长度、花萼宽度、花瓣（Petal）长度和花瓣宽度；包含一列分类信息，取值为亚属的名称（Iris setosa, Iris virginica and Iris versicolor）。该数据集是由 Edgar Anderson 在加拿大加斯帕半岛测量收集；Fisher, R.A. 在 1936 年以此数据集为例演示判别分析方法。

为了给大家一些直观的印象，列出了这三种鸢尾花的照片，如图 6-1、图 6-2 和图 6-3 所示。读者想看更清晰的图片及了解更多内容，可以访问网址：http://en.wikipedia.org/wiki/Iris_flower_data_set。



图 6-1　山鸢尾（Iris setosa）

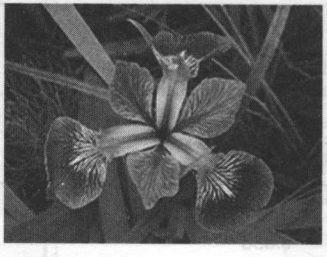


图 6-2　变色鸢尾（Iris virginica）

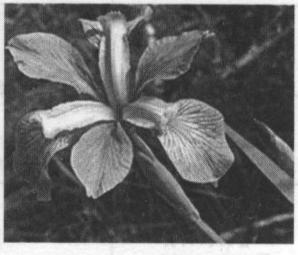


图 6-3　维吉尼亚鸢尾（Iris versicolor）

数据存在表 iris_data 中，内容显示如图 6-4 所示。

sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5.0	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa

图 6-4 数据表 iris_data 的内容

6.1 数据可视化

对于特征都为数值的样本，散点图可以展示不同特征所构成的二维平面空间上样本的分布情况，如图 6-5 所示，添加散点图组件。

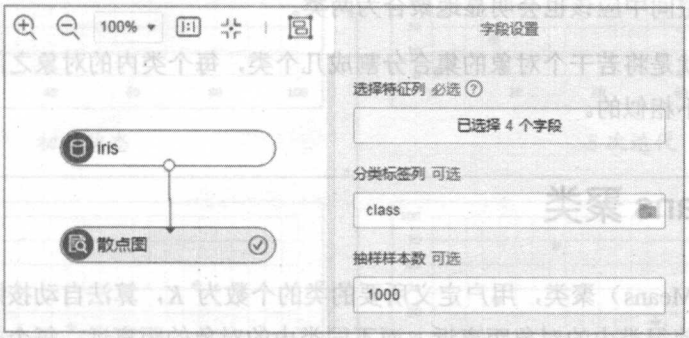


图 6-5 散点图组件的连接和字段设置

散点图组件运行结果，如图 6-6 所示，对角线上为各参与计算的特征列的名称，不同的特征分别作为横轴与纵轴，便可得二位散点图构成的矩阵。

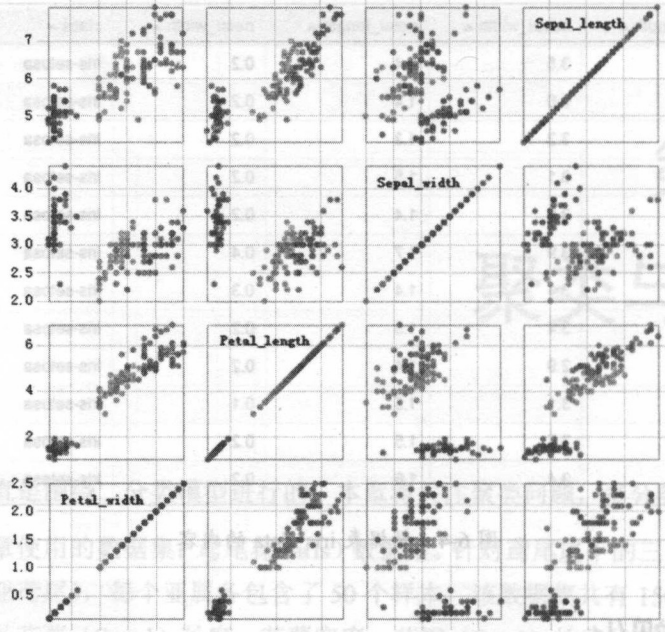


图 6-6　散点图组件运行结果

这些数据在每个散点图中都会聚为两簇，每个数据有 4 列属性，可以看作四维空间中的点，我们没法显示出数据在四维空间的分布，但是从数据在很多二维平面上的投影的情况，我们会猜想，其在四维空间中应该也会明显地聚合为两类。

聚类算法，就是将若干个对象的集合分割成几个类，每个类内的对象之间是相似的，但与其他类的对象是不相似的。

6.2　K-Means 聚类

K 均值（K-Means）聚类，用户定义所要的类的个数为 K，算法自动按照距离将数据自动聚合为 K 个类，使得类内的对象距离近，而不同类内的对象的距离远；每个类以该类所有数据的“均值”作为其中心点。

算法描述如下。

1）在给定的 n 个对象中，随机选取 k 个对象，作为每一个类的中心（初始均值），或者通过其他方式指定 k 个中心；

2）对于全部 n 个对象，分别计算与 k 个中心的距离，将对象指派到最接近的类；

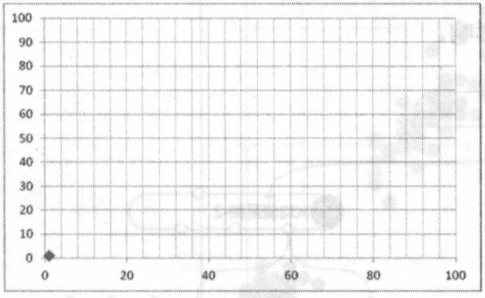
3）更新每个类的新均值，得出 k 个新的中心；

4）根据 k 个新的中心，重复 1）和 3）中的两个步骤，直至满足收敛准则。

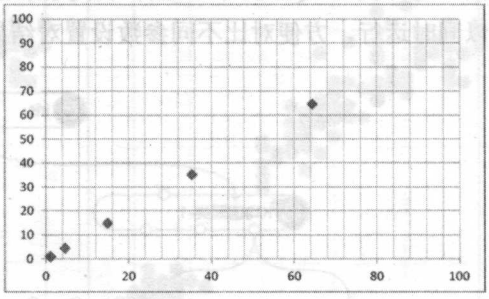
下面希望通过一个例子，使大家可以直观地理解。我们生成均匀分布的二维数据，每个维度的取值范围都是[0， 100]，生成 100 亿个这样的数据点。聚类的个数设为 9，我们可以想象到最佳的聚类状态：将整个区域按“井”字等分成 9 份。

下面通过 K-Means 聚类算法实际计算一下，这里假设了最坏的初始点情况，九点重合且位于区域的边缘，如表 6-1 的左上图所示，随后展现了 5 次迭代和 10 次迭代后的情况，我们惊奇地发现，重合的初始中心点被展开了；第 30 次迭代时，中心点已在整个区域散开了；到了第 50 次迭代的时候，这些中心点按 3×3 的位置排列；第 100 次迭代时，中心点已经排列得非常整齐了。

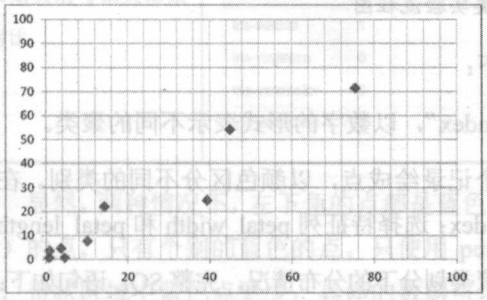
表 6-1 中心点位置随迭代次数的变化



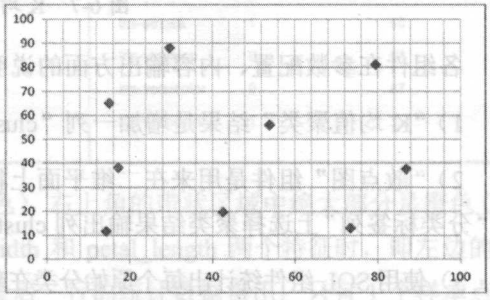
初始状态



5 次迭代

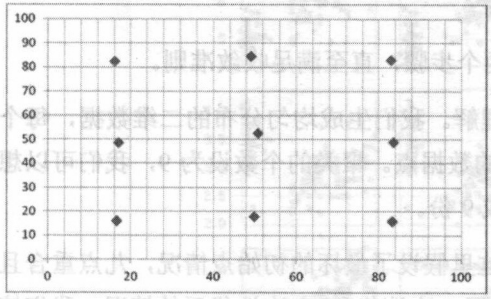


10 次迭代

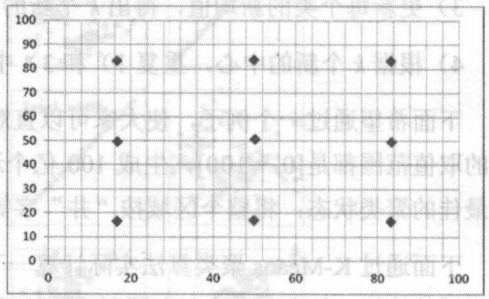


30 次迭代

续表



50 次迭代



100 次迭代

6.2.1 聚类、评估流程

图 6-7 展示了完整的聚类、评估流程。使用“K 均值聚类”组件进行聚类操作，并通过“散点图”组件及“SQL 脚本”组件对聚类的结果进行评估，注意这里列了两组聚类评估的组件，可以同时运行，方便对比不同参数设置对结果的影响。

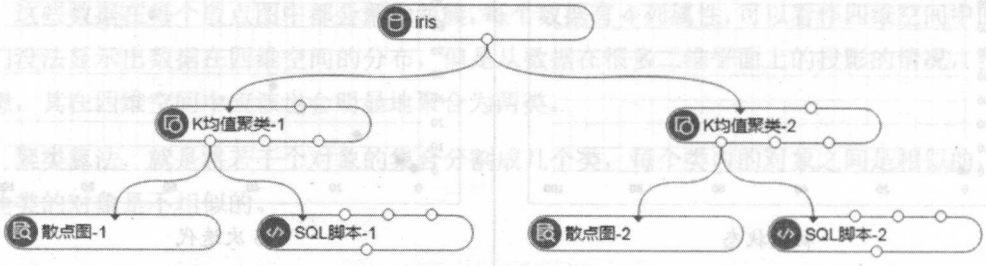


图 6-7 K 均值聚类实验流程图

各组件在参数配置、内容输出方面的说明如下：

1）“K 均值聚类”结果是增加一列“cluster_index”，以数字的形式表示不同的聚类。

2）“散点图”组件是用来在二维平面上将各个记录绘成点，以颜色区分不同的类别。在属性“分类标签列”上选择聚类结果输出列 cluster_index；选择特征列 petal_width 和 petal_length。

3）使用 SQL 组件统计出每个原始分类在新的聚类划分下的分布情况，完整 SQL 语句如下：

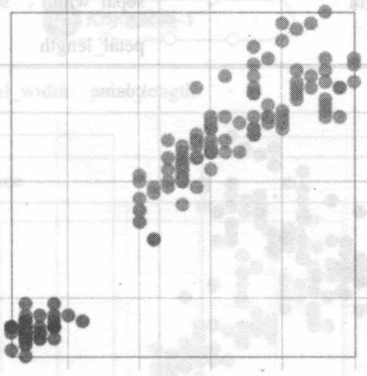
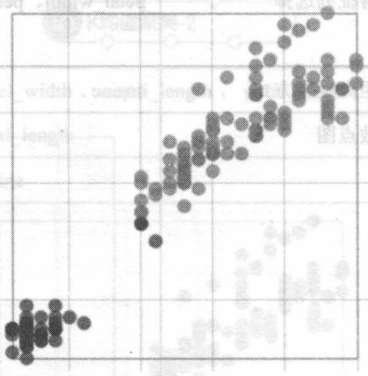
```
SELECT class, cluster_index, COUNT(class) as cnt FROM ${t1} GROUP BY class,
cluster_index
```

6.2.2 聚成两类

从图 6-6 的散点图上可以看出，在特征 petal_width 和 petal_length 所构成的二维空间中，样本在左下角聚集成为一簇，在右上角呈带状集中分布。所以，我们想先尝试聚类数为 2 的情形。

在实验参数的选择上，对于距离度量方式，先选择我们比较熟悉的欧式（Euclidean）距离；由图 6-6 的散点图可以看出，特征 petal_width 和 petal_length 比较特殊，可以将这两个特征作为一种情况，另外一种情况是选择全部 4 个特征列。具体的聚类参数选择及实验结果如表 6-2 所示。

表 6-2 欧式距离下的 K 均值聚类组件效果对比

	K均值聚类-1	K均值聚类-2
特征列选择	petal_width、petal_length	sepal_width、sepal_length、petal_width、petal_length
距离度量方式	Euclidean	Euclidean
散点图		
原始类别与聚类结果对比	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 1 50 Iris-virginica 0 50 Iris-versicolor 0 49 Iris-versicolor 1 1	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 1 50 Iris-virginica 0 50 Iris-versicolor 0 47 Iris-versicolor 1 3

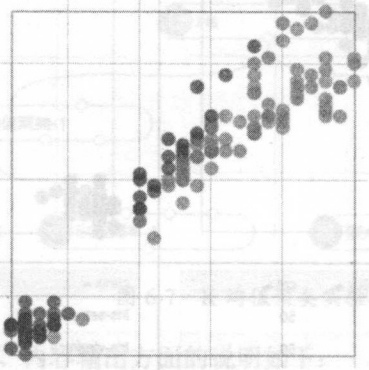
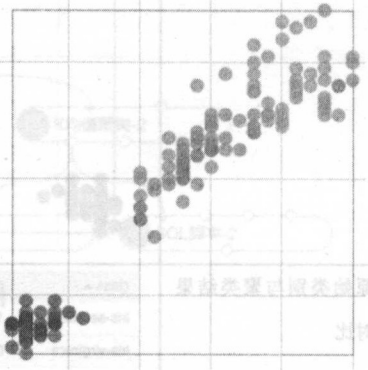
显然，两种情况下，左下角的点都是蓝色（深色），右上角的带状区域中绝大部分是橙色（浅色）的点，只有个别的蓝色的点。只使用 petal_width 和 petal_length 两个特征时，即左边的情况，带状区域中蓝色点更少，具体的数据可以由最后一行的统计数据得出：只有 1 个样本点。至于这个点为什么会分错，我们可以分析下原因：判断属于哪个类，是由该点到不同类中心的距离决定的，分错的那个点，虽然周围就有橙色点，但从距离上来讲，离蓝色点中心的距离却

更近。

K 均值聚类算法中的距离是有多种选择的，除了常用的欧式（Euclidean）距离，还有街区距离（CityBlock，对应向量的绝对误差之和）和余弦距离（cosine，对应向量的夹角余弦值）。在其他距离定义下，会有更好的结果吗？

下面我们实验余弦距离，从另外一个角度来区分数据，修改两个“K 均值聚类”组件的参数如表 6-3 所示，主要是修改距离度量方式为余弦（cosine）距离。从散点图分布上，选择 4 个特征列的效果很好，图中左下角的点都是同一种颜色，右上角呈带状分布的点也是同样的颜色；我们再从样本原始标签与聚类索引号的统计结果中，进行量化的分析，Iris setosa 分类的样本都属于索引号为 1 的聚类，Iris virginica 和 Iris versicolor 分类的样本都属于索引号为 0 的聚类。

表 6-3　余弦距离下的 K 均值聚类组件效果对比

	K均值聚类-1	K均值聚类-2
特征列选择	petal_width、petal_length	sepal_width 、 sepal_length 、 petal_width 、 petal_length
距离度量方式	cosine	cosine
散点图		
原始类别与聚类结果对比	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 0 2 Iris-setosa 1 48 Iris-virginica 0 46 Iris-virginica 1 4 Iris-versicolor 0 36 Iris-versicolor 1 14	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 1 50 Iris-virginica 0 50 Iris-versicolor 0 50

最后，我们再将欧式距离与余弦距离下的两组试验，即表 6-2 和表 6-3 的数据，放在一起

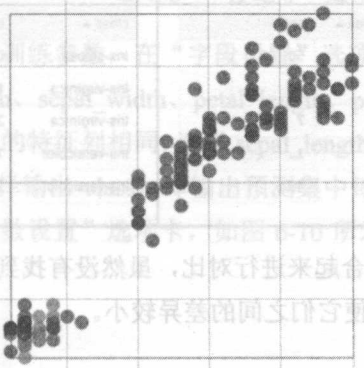
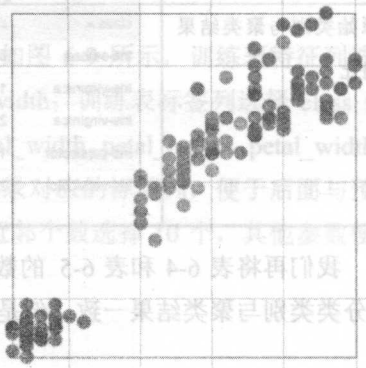
分析，会发现最初我们凭直觉选择的两个特征 petal_width 和 petal_length 在欧式距离定义下，聚类效果不错；但是最好的聚类划分却是在余弦距离下和全部特征参加的情况下取得的。相信经过这个实验，我们对聚类特征的选择，距离定义的选择都会有新的心得。

6.2.3　聚成三类

有了上节的基础，我们来挑战更有难度的聚类数为 3 的情形。原始的样本数据本来就属于三个分类，但是由图 6-6 的散点图可以看出，Iris virginica 和 Iris versicolor 分类的样本点没有清晰的分界线。如何能较好地聚成三簇，还尽量与其原来的分类属性一致？

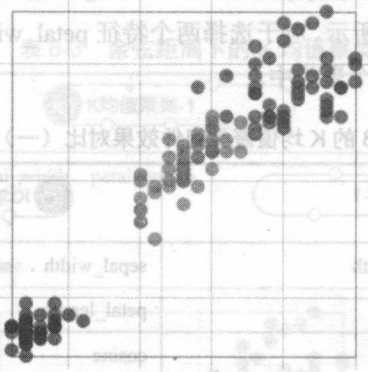
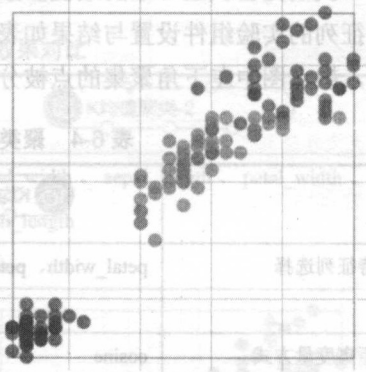
我们先尝试上一节的最佳参数组合，在余弦（cosine）距离下，选择两个特征列与选择全部特征列的实验组件设置与结果如表 6-4 所示。对于选择两个特征 petal_width 和 petal_length 的情况，散点图中左下角聚集的点被分到三个聚类中。

表 6-4　聚类数为 3 的 K 均值聚类组件效果对比（一）

	K均值聚类-1	K均值聚类-2
特征列选择	petal_width、petal_length	sepal_width 、 sepal_length 、 petal_width 、 petal_length
距离度量方式	cosine	cosine
散点图		
原始类别与聚类结果对比	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 0 7 Iris-setosa 1 16 Iris-setosa 2 27 Iris-virginica 0 50 Iris-versicolor 0 50	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 2 50 Iris-virginica 0 47 Iris-virginica 1 3 Iris-versicolor 0 3 Iris-versicolor 1 47

下面尝试欧式距离下的实验，选择两个特征列与选择全部特征列的实验组件设置与结果如表 6-5 所示。从散点图上不容易看出区别，通过对比原始分类类别与聚类结果的统计数据，对于选择两个特征列的情况会更好一些。

表 6-5　聚类数为 3 的 K 均值聚类组件效果对比（二）

	K均值聚类-1	K均值聚类-2
特征列选择	petal_width、petal_length	sepal_width、sepal_length、petal_width、petal_length
距离度量方式	Euclidean	Euclidean
散点图		
原始类别与聚类结果对比	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 0 50 Iris-virginica 1 43 Iris-virginica 2 7 Iris-versicolor 1 1 Iris-versicolor 2 49	class ▲ / cluster_index ▲ / cnt ▲ Iris-setosa 0 50 Iris-virginica 1 36 Iris-virginica 2 14 Iris-versicolor 1 2 Iris-versicolor 2 48

我们再将表 6-4 和表 6-5 的数据综合起来进行对比，虽然没有找到一种参数设置，使原始的分类类别与聚类结果一致，但是可以使它们之间的差异较小。

6.3　K 最近邻算法

K 最近邻算法（K-Nearest Neighbor，缩写为 KNN）：对某个样本分类情况的预测，可以用与其最接近的 k 个邻居的类别情况来判断。

该算法所遵循的准则是“近朱者赤，近墨者黑”，判断一个人的好坏，通过已知的与其关系最近的K个人的情况来判断，如果大多数为好人，则判断此人为好人，反之，则认为其为坏人。

KNN算法虽然很简单，但在某些场景下效果还是可以的，比如在本节正在讨论的iris数据的分类场景。

6.3.1 使用KNN算法进行分类

首先将iris数据表连接到“拆分”组件，并设置切分比例为0.8，即左、右输出桩输出数据的比例为0.8。加入“K近邻”组件，左边的输入桩连接训练数据，右边的输入桩连接预测数据，如图6-8所示。

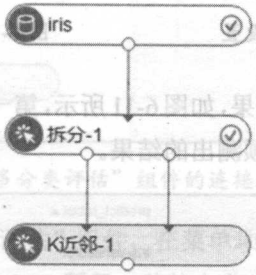


图6-8 “K近邻”组件的连接

然后再进一步设置训练参数，在“字段设置”选项卡，如图6-9所示，训练表特征列选择所有数值列sepal_length、sepal_width、petal_length、petal_width；训练表标签列选择class列；预测表特征列与训练表的特征列相同，选择sepal_length、sepal_width、petal_length、petal_width；产出表附加ID列：选择输出class，即输出预测集中每条记录对应的标签列，便于后面与预测结果进行对比。在“参数设置”选项卡，如图6-10所示，近邻个数选择10个，其他参数使用默认值即可。

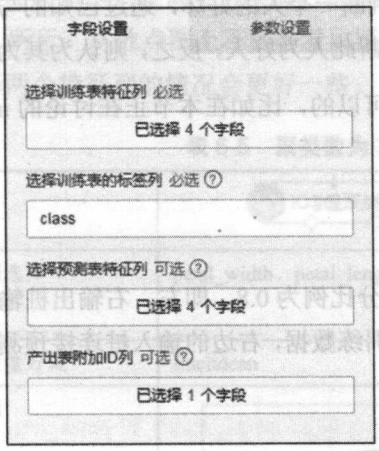


图 6-9 “K 近邻”组件的字段设置

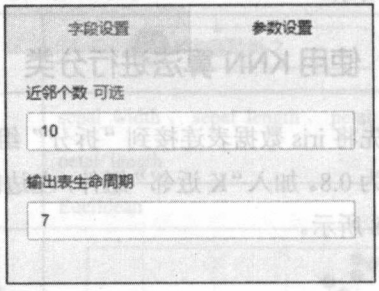


图 6-10 “K 近邻”组件的参数设置

K 近邻组件的输出为分类预测结果，如图 6-11 所示，第一列是预测表保留下来的标签列 class，第二列 predict_label 是由 KNN 算法预测出的结果。

class	predict_label
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-virginica	Iris-virginica
Iris-virginica	Iris-virginica
Iris-virginica	Iris-versicolor
Iris-virginica	Iris-virginica
Iris-virginica	Iris-virginica
Iris-virginica	Iris-virginica
Iris-virginica	Iris-virginica

图 6-11 “K 近邻”组件的计算结果

我们可以看到，预测的结果与原始标签大部分还是一致的。如果想要拿到量化的评估指标，可以使用“多分类评估”组件，并设置原分类结果列为 class，预测分类结果列为 predict_label，如图 6-12 所示。

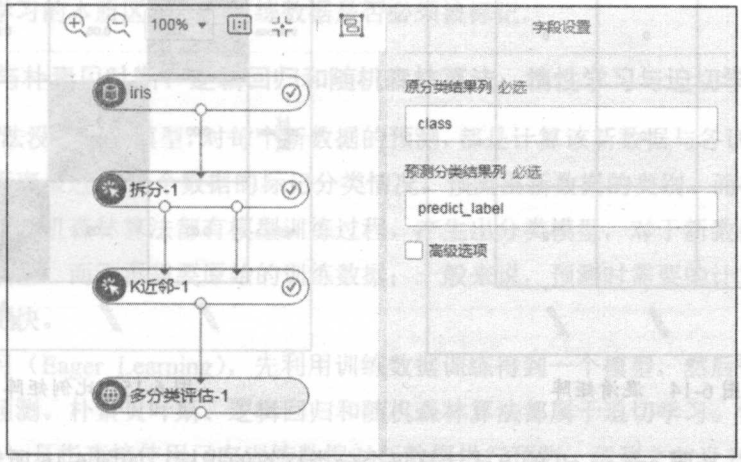


图 6-12　“多分类评估”组件的连接和字段设置

待计算完毕，右键点击“多分类评估”组件，在菜单项中选择“查看评估报告”，可以得到多分类评估指标的可视化展示，如图 6-13 所示，总体上分为四个选项卡，下面将一一介绍。在“总览”项中列出了两个最重要的指标：精确度（Accuracy）和 Kappa 值。

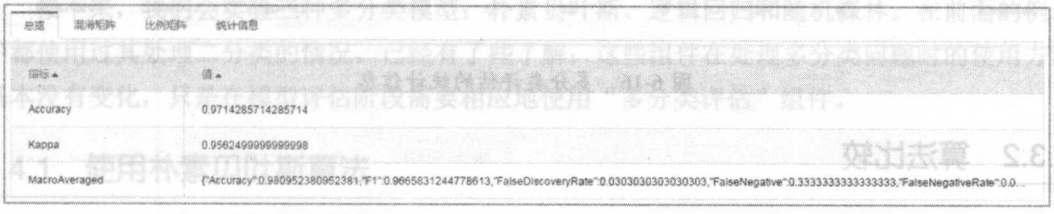


图 6-13　“多分类评估”组件的评估报告

通过“混淆矩阵”与“比例矩阵”选项卡，如图 6-14 和图 6-15 所示，我们可以了解更多细节。混淆矩阵对角线上的各个单元代表着原始标签值与预测值相同的情况，我们看到预测对了 15+10+9=34 个样本，预测错误的只有一个样本（真实值是 iris-virginica，预测成了 iris-versicolor）。比例矩阵，是在混淆矩阵的基础上，把每个单元的值除以每列各单元的和。

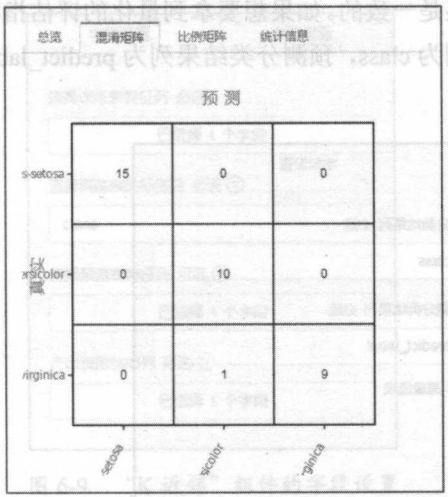


图 6-14　混淆矩阵

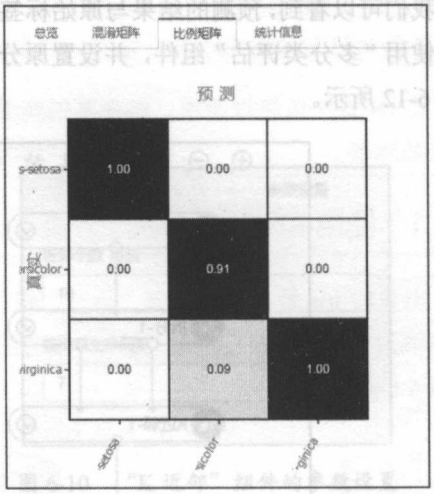


图 6-15　比例矩阵

“统计信息”按照 one-vs-all 的方式计算每个分类标签的指标，如图 6-16 所示。

总览　混淆矩阵　比例矩阵　统计信息

模型	TruePositive	TrueNegative	FalsePositive	FalseNegative	Sensitivity	Specificity	Precision	Accuracy	F1	Kappa
Iris-setosa	15	20	0	0	1	1	1	1	1	1
Iris-versicolor	10	24	1	0	1	0.96	0.90909090...	0.971428571...	0.952380952...	0.932038834...
Iris-virginica	9	25	0	1	0.9	1	1	0.971428571...	0.947368421...	0.927835051...

图 6-16　多分类评估的统计信息

6.3.2　算法比较

我们还可以通过将 KNN 算法与其他算法进行比较，帮助读者理解机器学习的一些概念，并更深入地了解这些算法的本质。

1. KNN 算法与 K-Means 算法；监督学习与非监督学习

- KNN 算法的训练数据标记了每条记录的类别信息，但 K-Means 的训练数据不需要标记类别信息。

- 监督学习（Supervised Learning）是从对标记的训练数据进行学习。KNN、朴素贝叶斯、逻辑回归、随机森林等分类方法都属于监督学习。非监督学习（Unsupervised Learning）是对没有标记的训练数据进行学习。典型的例子就是 K-Means 等聚类方法。监督学习与非监督学习的本质区别在于训练数据是否必须被标记。

2. KNN 算法与朴素贝叶斯、逻辑回归和随机森林算法；惰性学习与迫切学习

- KNN 算法没有产生模型，对每个新数据的预测，都是计算该新数据与各训练数据的距离，再根据距离最近的 K 个数据的标记分类情况，预测出新数据的类别。而朴素贝叶斯、逻辑回归和随机森林算法都有模型训练过程，产生出分类模型，对于新数据的预测只需要有分类模型，而不再需要原始的训练数据；一般来说，预测时需要的计算量更小，预测的速度更快。
- 迫切学习（Eager Learning），先利用训练数据训练得到一个模型，然后使用模型对新数据进行预测。朴素贝叶斯、逻辑回归和随机森林算法都属于迫切学习。惰性学习（Lazy Learning）是指直接使用已有训练数据对新数据进行预测，预测之前没有模型训练过程，KNN 属于这种方式。

6.4 多分类模型

接下来，我们会实验三种多分类模型：朴素贝叶斯、逻辑回归和随机森林。在前面的例子中都使用过其处理二分类的情况，已经有了些了解；这些组件在处理多分类问题时的使用方法基本没有变化，只是在模型评估阶段需要相应地使用“多分类评估”组件。

6.4.1 使用朴素贝叶斯算法

添加“朴素贝叶斯”组件，训练数据连接唯一的输入桩，设置其参数，特征列选择所有数值列 sepal_length，sepal_width，petal_length，petal_width；标签列选择 class。如图 6-17 所示。

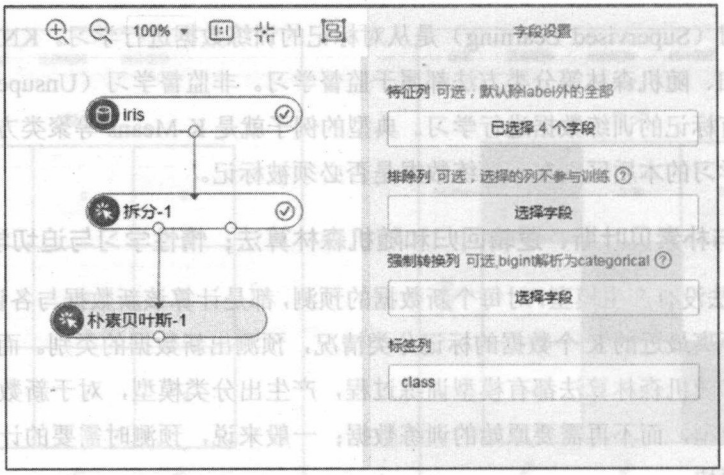


图 6-17 “朴素贝叶斯”组件的连接及字段设置

通过“朴素贝叶斯”组件，我们可以得到模型，接下来使用“预测”组件，将模型应用到具体的预测数据上，如图 6-18 所示。“预测”组件的特征列，选择所有数值列 sepal_length，sepal_width，petal_length，petal_width；原样输出的列选择标签列 class，后面的评估需要参考此值；其他的参数值可以直接使用系统提供的默认值。

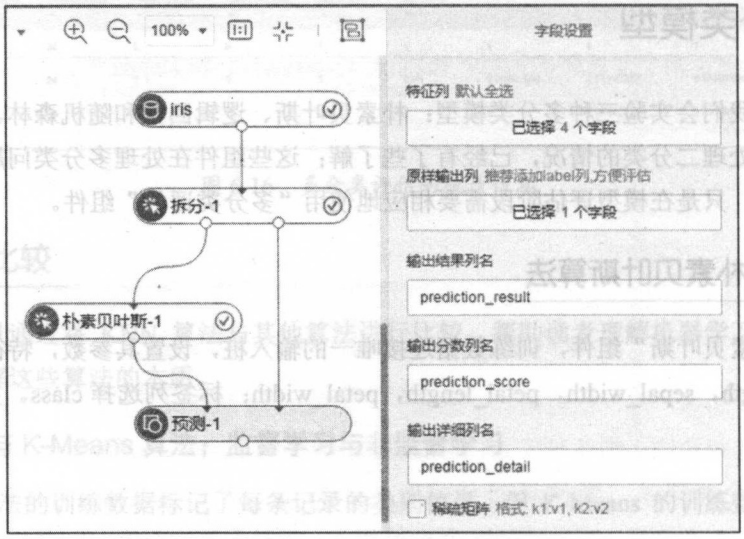


图 6-18 “预测”组件的连接及字段设置

计算完成后，查看预测结果。如图 6-19 所示，其中第一列 class 为原始标签，第二列 prediction_result 为预测的分类结果。

class	prediction_result	prediction_score	prediction_detail
Iris-versicolor	Iris-versicolor	-2.2345408966551616	{"Iris-setosa": -173.0390093037547, "Iris-versicolor": -2.234540896655162, "Iris-virginica": -6.507066256615971}
Iris-versicolor	Iris-versicolor	-2.299947890281252	{"Iris-setosa": -108.251203883657, "Iris-versicolor": -2.299947890281252, "Iris-virginica": -15.16130455412947}
Iris-versicolor	Iris-versicolor	-1.3193233677383676	{"Iris-setosa": -176.7045622877255, "Iris-versicolor": -1.319323367738368, "Iris-virginica": -8.08833821078729}
Iris-versicolor	Iris-versicolor	-1.1560312281143883	{"Iris-setosa": -154.3257183804258, "Iris-versicolor": -1.156031228114388, "Iris-virginica": -7.949094497858622}
Iris-versicolor	Iris-versicolor	-3.6180886946744018	{"Iris-setosa": -78.66562585352516, "Iris-versicolor": -3.618088694674402, "Iris-virginica": -18.60394786242802}
Iris-versicolor	Iris-versicolor	-1.271395533233663	{"Iris-setosa": -133.2705404325536, "Iris-versicolor": -1.271395533233663, "Iris-virginica": -10.36287980361291}
Iris-versicolor	Iris-versicolor	-1.6562653318225404	{"Iris-setosa": -128.1967900179762, "Iris-versicolor": -1.65626533182254, "Iris-virginica": -12.34087895767916}
Iris-versicolor	Iris-versicolor	-0.9117853823051482	{"Iris-setosa": -152.2901711400144, "Iris-versicolor": -0.9117853823051482, "Iris-virginica": -8.13454385591853}
Iris-virginica	Iris-virginica	-1.618920592901239	{"Iris-setosa": -402.0036066253426, "Iris-versicolor": -15.02810168751658, "Iris-virginica": -1.618920592901239}
Iris-virginica	Iris-virginica	-4.498540132971835	{"Iris-setosa": -499.9880479516046, "Iris-versicolor": -23.09231177722795, "Iris-virginica": -4.498540132971835}
Iris-virginica	Iris-versicolor	-4.6102128468739965	{"Iris-setosa": -200.5546829928109, "Iris-versicolor": -4.610212846873996, "Iris-virginica": -9.259920672750876}
Iris-virginica	Iris-virginica	-3.5261151992355115	{"Iris-setosa": -345.9883669520675, "Iris-versicolor": -14.77612148719518, "Iris-virginica": -3.526115199235512}
Iris-virginica	Iris-virginica	-9.232660925187043	{"Iris-setosa": -529.372247477659, "Iris-versicolor": -31.06943012692995, "Iris-virginica": -9.232660925187043}
Iris-virginica	Iris-versicolor	-2.527555071493124	{"Iris-setosa": -238.9382726624566, "Iris-versicolor": -2.527555071493124, "Iris-virginica": -3.771164474016163}
Iris-virginica	Iris-versicolor	-4.407419907295516	{"Iris-setosa": -280.7707415791793, "Iris-versicolor": -4.407419907295516, "Iris-virginica": -5.121062738858906}
Iris-virginica	Iris-virginica	-3.2714232113347377	{"Iris-setosa": -400.2356395486876, "Iris-versicolor": -18.86289930609725, "Iris-virginica": -3.271423211334738}
Iris-virginica	Iris-virginica	-1.295455648347823	{"Iris-setosa": -343.1644426269456, "Iris-versicolor": -11.77223551342525, "Iris-virginica": -1.295455648347823}
Iris-virginica	Iris-virginica	-2.910471256973107	{"Iris-setosa": -362.6128545886669, "Iris-versicolor": -15.48899227387414, "Iris-virginica": -2.910471256973107}

图 6-19 多分类预测结果

使用“多分类评估”组件，并设置原分类结果列为 class，预测分类结果列为 prediction_result，如图 6-20 所示。

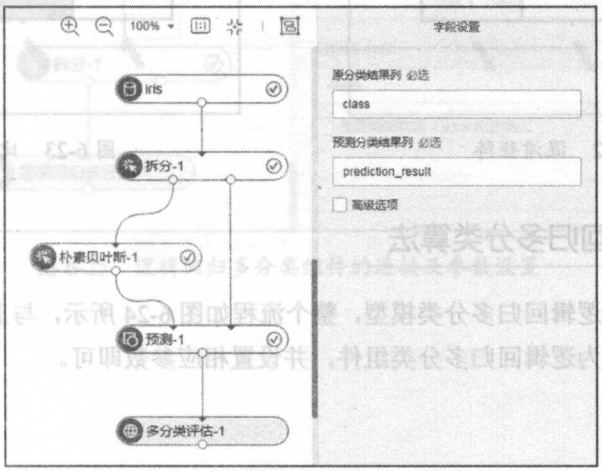


图 6-20 “多分类评估”组件的连接和字段设置

多分类评估的主要评估指标如图 6-21、图 6-22 和图 6-23 所示。

总览　混淆矩阵　比例矩阵　统计信息

指标	值
Accuracy	0.9142857142857143
Kappa	0.8687499999999998
MacroAveraged	{"Accuracy":0.9428571428571427,"F1":0.8976982097186701,"FalseDiscoveryRate":0.07692307692307693,"FalseNegative":1,"FalseNegative...

图 6-21　多分类评估的指标总览

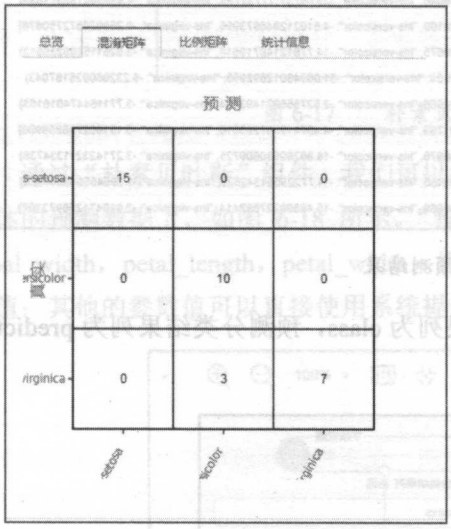


图 6-22　混淆矩阵

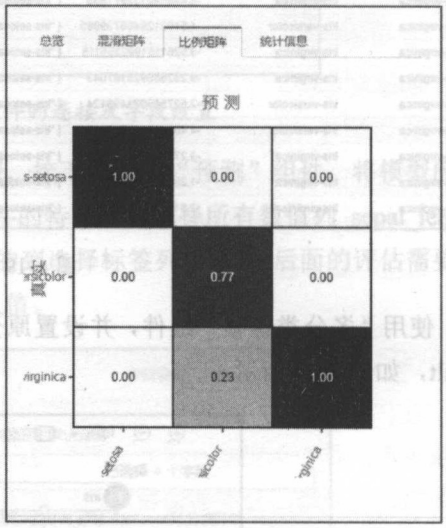


图 6-23　比例矩阵

6.4.2　使用逻辑回归多分类算法

本节我们将尝试逻辑回归多分类模型，整个流程如图 6-24 所示，与上节非常相似，只要将朴素贝叶斯组件替换为逻辑回归多分类组件，并设置相应参数即可。



图 6-24　逻辑回归多分类模型

“逻辑回归多分类”组件的字段设置如图 6-25 所示，特征列选择所有数值列 sepal_length，sepal_width，petal_length，petal_width；标签列选择 class。

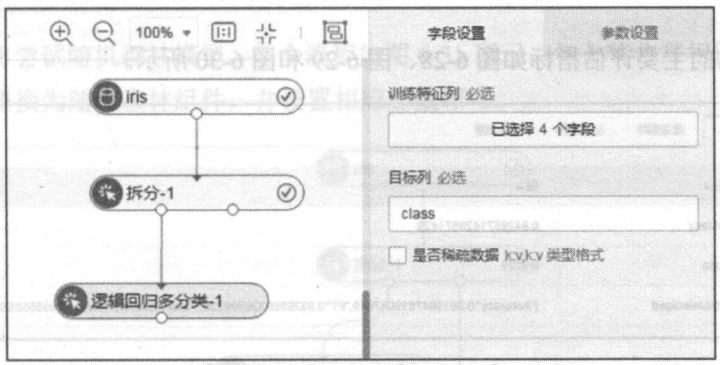


图 6-25　逻辑回归多分类组件的连接及参数设置

计算完成后，我们可以在最左端黑色区域选择“模型”图标，可以看到“实验模型”，选择本次实验名称的文件夹，即可看到刚才计算出的逻辑回归多分类模型，再点击右键，选择“查看模型”，如图 6-26 所示。得到的模型描述如图 6-27 所示。



图 6-26　查看模型

逻辑回归输出

字段名	权重		
	Iris-setosa	Iris-versicolor	Iris-virginica
sepal_length	3.196020009121547	-0.83528825111013...	4.365520119195724
sepal_width	10.99140661566953	-2.196479526328716	-267.0973498305601
petal_length	-16.57856128629576	1.903069242466502	163.9833774067979
petal_width	-8.078063915707759	-3.571114109266009	796.6330595751378
常量	0.862052023048024	7.808055077467728	-1430.597771432667

1、PAI平台提供的逻辑回归可用于多分类的，采取的策略是OneVsAll，因此在多分类的情况下，会出现多个方程，每个方程针对目标特征的某个value值，即权重（weight）下方对应的列名；

2、逻辑回归的完整公式为：σ(z) = 1 / (1 + exp(z))；z = w0 + w1 * x1 + w2 * x2 + ... + wm * xm。（其中x1, x2, ..., xm 是某样本数据的各个特征，w1, w2, ... 是特征的权重值）

关闭

图 6-27　逻辑回归多分类模型的参数

多分类评估的主要评估指标如图 6-28、图 6-29 和图 6-30 所示。

总览　混淆矩阵　比例矩阵　统计信息

指标	值
Accuracy	0.9428571428571428
Kappa	0.9125
MacroAveraged	{"Accuracy":0.9619047619047619,"F1":0.9326599326599326,"FalseDiscoveryRate":0.0555555555555...

图 6-28　多分类评估结果总览

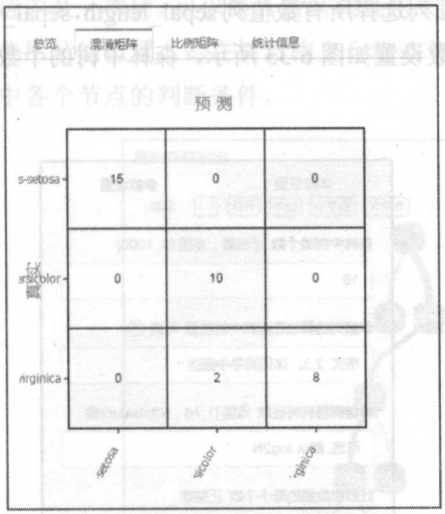


图 6-29 多分类评估结果之混淆矩阵

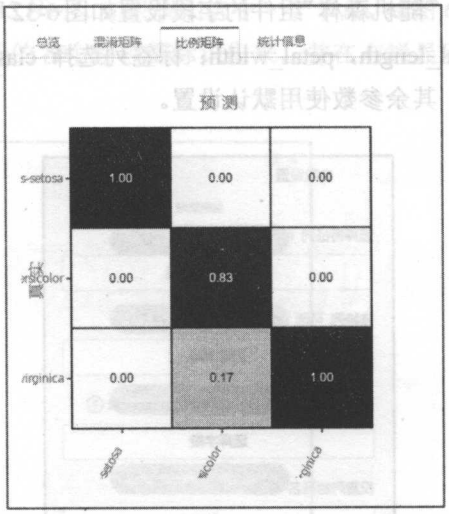


图 6-30 多分类评估结果之比例矩阵

6.4.3 使用随机森林算法

本节我们将尝试随机森林模型，整个流程如图 6-31 所示，与上节非常相似，只要将逻辑回归多分类组件替换为随机森林组件，并设置相应参数即可。



图 6-31 随机森林模型

“随机森林”组件的字段设置如图6-32所示，特征列选择所有数值列sepal_length，sepal_width，petal_length，petal_width；标签列选择 class。其参数设置如图 6-33 所示，森林中树的个数选择10，其余参数使用默认设置。

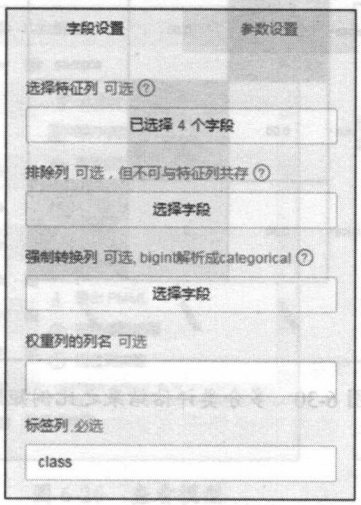


图 6-32　随机森林组件的字段设置

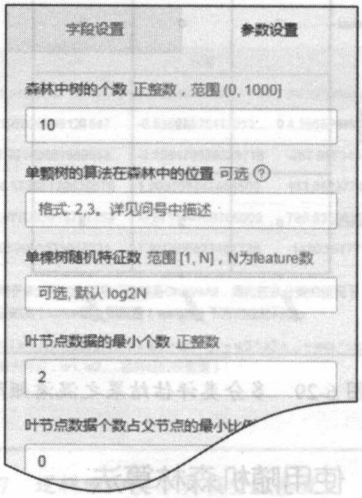


图 6-33　随机森林组件的参数设置

随机森林模型输出的结果为若干棵决策树构成的“森林”，如图 6-34 所示。

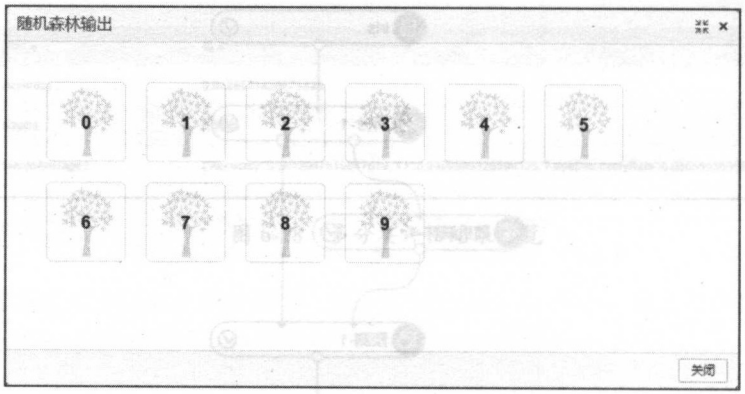


图 6-34　随机森林模型结果

选中某一棵树，可以将其树结构画出来，如图 6-35 所示。如果我们选中一个节点，如图 6-35 上箭头所示的位置，则在树结构上会将该节点所经历的各个路径都标识出来，并在右栏显示出途径中各个节点的判断条件。

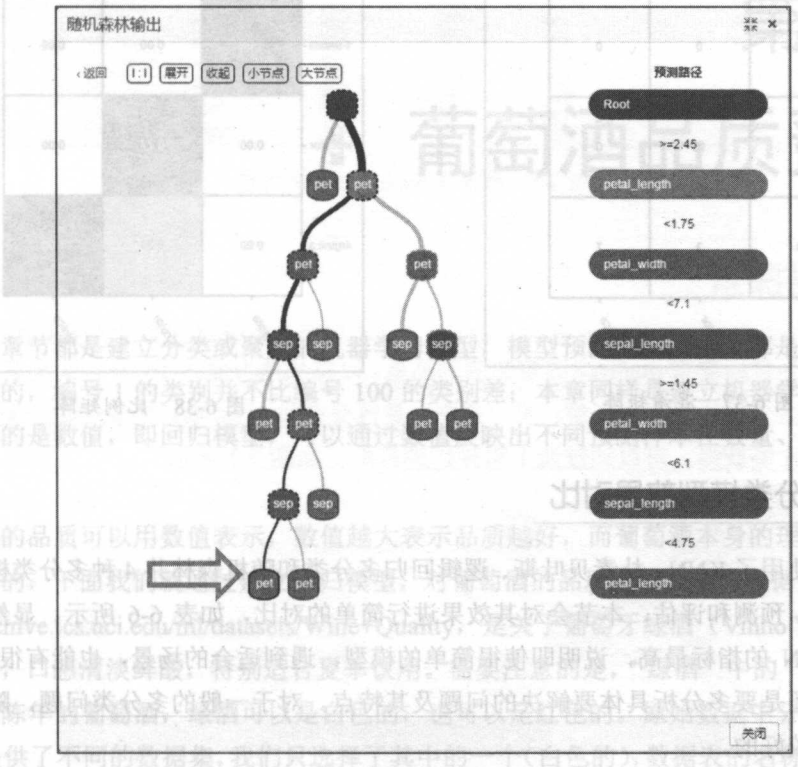


图 6-35　随机森林模型中的决策树展示

随机森林模型多分类评估的主要评估指标如图 6-36、图 6-37 和图 6-38 所示。

总览　混淆矩阵　比例矩阵　统计信息

指标	值
Accuracy	0.9142857142857143
Kappa	0.8687499999999998
MacroAveraged	{"Accuracy":0.9428571428571427,"F1":0.8976982097186701,"FalseDiscoveryRate":0.076923...

图 6-36　多分类评估的指标总览

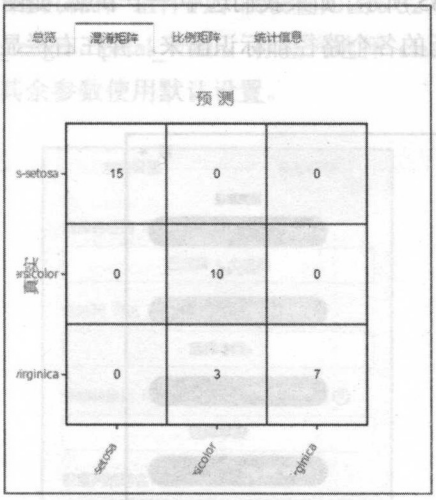


图 6-37　混淆矩阵

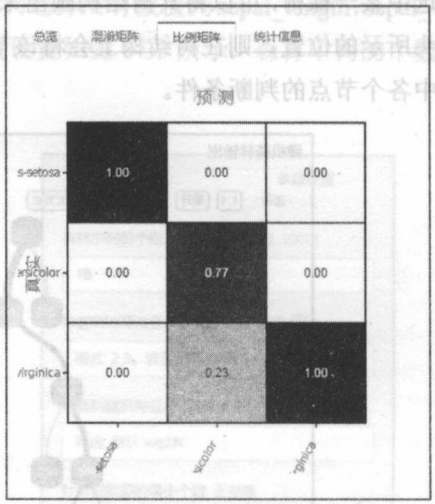


图 6-38　比例矩阵

6.4.4　各多分类模型效果对比

前面我们使用了 KNN、朴素贝叶斯、逻辑回归多分类和随机森林共 4 种多分类模型对此问题进行了建模、预测和评估，本节会对其效果进行简单的对比，如表 6-6 所示。显然，对于当前的场景，KNN 的指标最高，说明即使很简单的模型，遇到适合的场景，也能有很好的效果，我们在建模时还是要多分析具体要解决的问题及其特点。对于一般的多分类问题，随机森林模型的效果是比较好的。

表 6-6　四种多分类模型评估指标对比

模　型	精确度	Kappa 值
KNN	0.9714	0.9562
朴素贝叶斯	0.9143	0.8687
逻辑回归多分类	0.9429	0.9125
随机森林	0.9143	0.8687

第7章 葡萄酒品质预测

前面的章节都是建立分类或聚类的机器学习模型，模型预测输出的结果都是类别，所有的类别是平等的，编号 1 的类别并不比编号 100 的类别差；本章同样是建立机器学习模型，但模型预测输出的是数值，即回归模型，可以通过数值反映出不同预测样本在数量、品质、程度上的差异。

葡萄酒的品质可以用数值表示，数值越大表示品质越好，而葡萄酒本身的理化指标对其品质是有影响的，下面我们就通过建立回归模型，对葡萄酒的品质进行预测。数据来自 UCI 数据集 http://archive.ics.uci.edu/ml/datasets/Wine+Quality，是关于葡萄牙绿酒（Vinho Verde）的，其酒精度中等，口感清淡鲜酸，特别适合夏季饮用。需要注意的是，“绿酒”中的“绿”指的是还没有成熟或陈年的葡萄酒，绿酒可以是白色的，也可以是红色的。原始数据中分别对两种颜色的“绿酒”提供了不同的数据集，我们只选择了其中的一个（白色的），数据表的名称为 wine_white。

该数据集有 11 个特征变量，都是经过理化测试得到的数据，具体名称及含义如表 7-1 所示。

表 7-1　特征字段说明

编号	特征字段名称	中文含义	编号	特征字段名称	中文含义
1	fixed acidity	固定酸度	7	total sulfur dioxide	总二氧化硫
2	volatile acidity	挥发性酸度	8	density	密度
3	citric acid	柠檬酸	9	pH	pH 值
4	residual sugar	残糖	10	sulphates	硫酸盐
5	Chlorides	氯化物	11	alcohol	酒精
6	free sulfur dioxide	自由二氧化硫			

该数据集的目标变量为葡萄酒的品质，目标变量名称为 quality，分值在 0 和 10 之间，是基于感官的数据。显示数据片段如图 7-1 所示。

fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	ph	sulphates	alcohol	quality
6.6	0.17	0.38	1.5	0.032	28.0	112.0	0.9914	3.25	0.55	11.4	7.0
6.3	0.48	0.04	1.1	0.046	30.0	99.0	0.9928	3.24	0.36	9.6	6.0
6.2	0.66	0.48	1.2	0.029	29.0	75.0	0.9892	3.33	0.39	12.8	8.0
7.4	0.34	0.42	1.1	0.033	17.0	171.0	0.9917	3.12	0.53	11.3	6.0
6.5	0.31	0.14	7.5	0.044	34.0	133.0	0.9955	3.22	0.5	9.5	5.0
6.2	0.66	0.48	1.2	0.029	29.0	75.0	0.9892	3.33	0.39	12.8	8.0
6.4	0.31	0.38	2.9	0.038	19.0	102.0	0.9912	3.17	0.35	11.0	7.0

图 7-1　数据集 wine_white 的内容

7.1　数据探索

关注重点为目标变量 quality，即葡萄酒的品质。可以从两方面入手，一方面是其分布情况，可以通过计算该变量的直方图了解；另一方面，是该变量与 11 个特征变量间的关系，需要计算相关系数矩阵才能获得。可以通过两个组件达到目的，如图 7-2 所示。

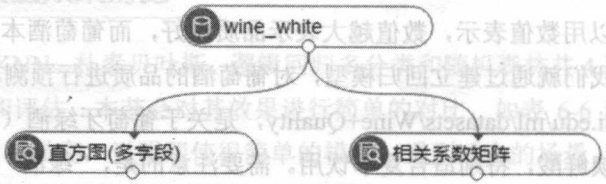


图 7-2　数据探索组件连接

其中“直方图（多字段）”组件中需要选择“quality”字段；“相关系数矩阵”组件不用新设置参数，默认计算全部。

字段 quality 的直方图如图 7-3 所示，可以知道 6 的个数最多；其次是 5 和 7；值 4 和 8 的分布更少，都在 200 以下，比值 6 的个数少一个数量级；而值 3 和 9 的比例更小。

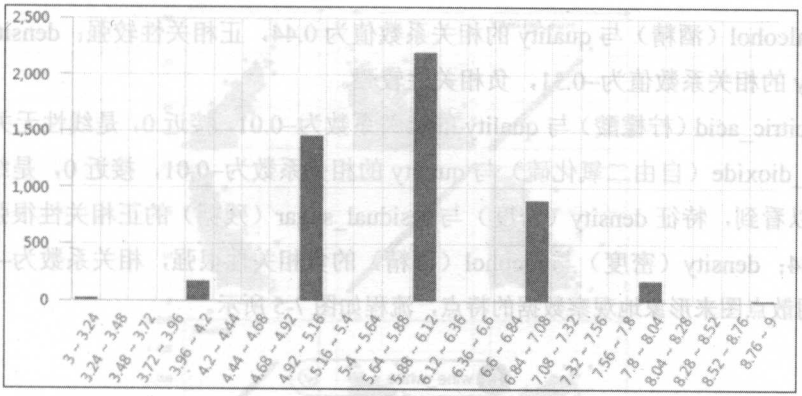


图 7-3　字段 quality 的直方图

再看各字段间的相关系数情况，如图 7-4 所示，除了有具体的数字，还用颜色进行了标识，蓝色（深色）为+1，橙色（浅色）为–1，0 用白色表示，颜色越淡说明线性相关性越弱，蓝色越深说明正相关性越强，橙色越深说明负相关性越强。

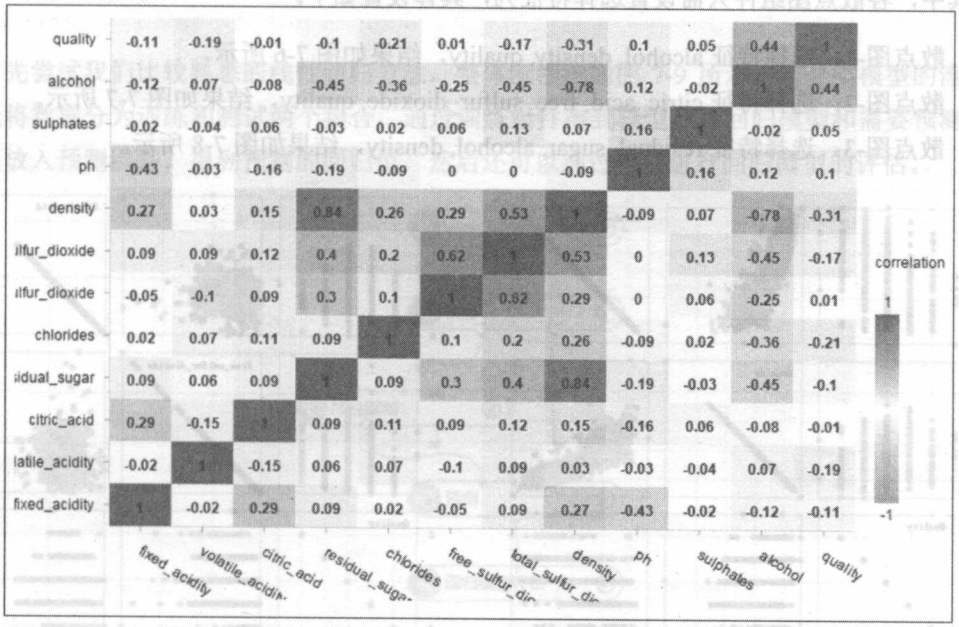


图 7-4　相关系数矩阵

从图 7-4 中可以看出如下特点。

- 特征 alcohol（酒精）与 quality 的相关系数值为 0.44，正相关性较强；density（密度）与 quality 的相关系数值为–0.31，负相关性较强。
- 特征 citric_acid（柠檬酸）与 quality 的相关系数为–0.01，接近 0，是线性无关；特征 free_sulfur_dioxide（自由二氧化硫）与 quality 的相关系数为–0.01，接近 0，是线性无关。
- 还可以看到，特征 density（密度）与 residual_sugar（残糖）的正相关性很强，相关系数为 0.84；density（密度）与 alcohol（酒精）的负相关性很强，相关系数为–0.78。

下面使用散点图来形象地观察数据的特点，流程如图 7-5 所示。

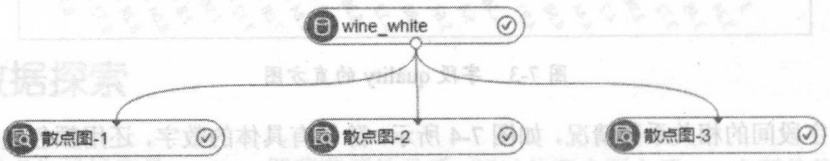


图 7-5　计算散点图

其中，各散点图组件只需设置选择特征列，具体设置如下。

- 散点图-1：选择特征 alcohol, density, quality，结果如图 7-6 所示。
- 散点图-2：选择特征 citric_acid, free_sulfur_dioxide, quality，结果如图 7-7 所示。
- 散点图-3：选择特征 residual_sugar, alcohol, density，结果如图 7-8 所示。

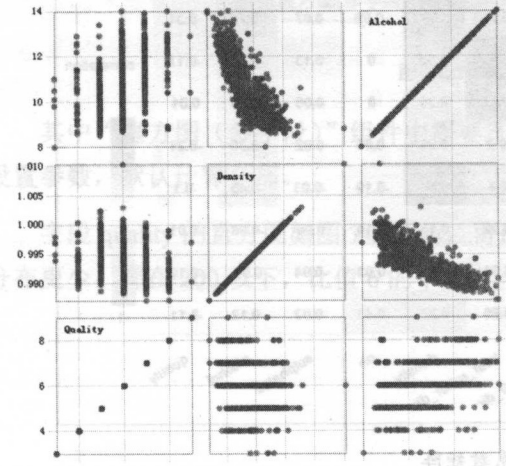


图 7-6　散点图-1

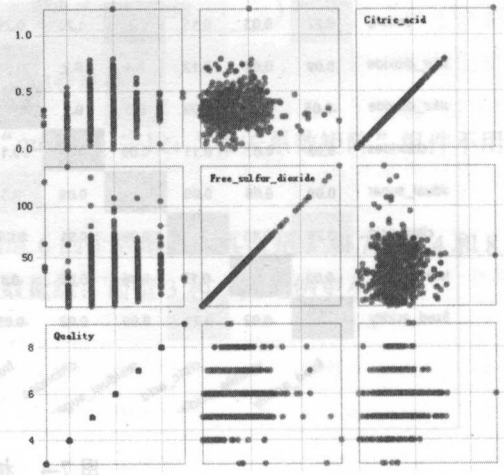


图 7-7　散点图-2

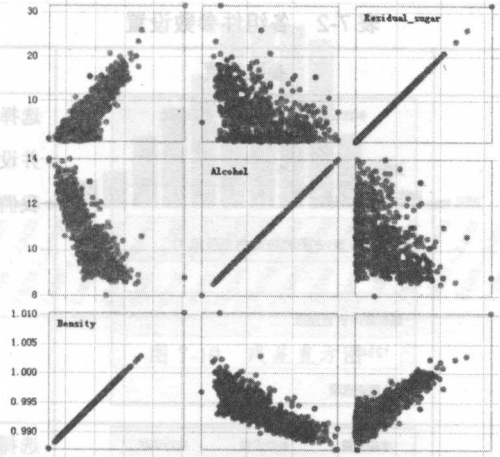


图 7-8　散点图-3

7.2　线性回归

首先尝试我们比较熟悉的线性回归方法，整体流程图如图 7-9 所示。与分类模型的流程相似，先将数据分为训练和测试两个集合，通过训练组件得到模型，将回归模型和需要预测的样本特征放入预测组件，得到预测的回归值，然后还可以通过组件进行回归模型的评估。

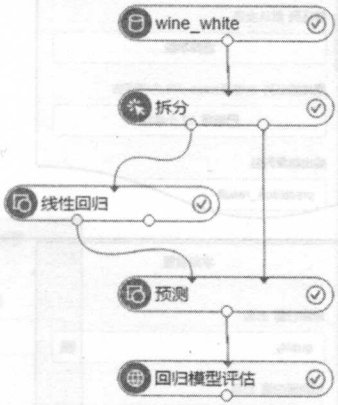


图 7-9　线性回归流程

其中各组件参数的设置如表 7-2 所示。

表 7-2　各组件参数设置

组　件	参数设置	说　明
拆分	参数设置　执行调优 拆分方式 按比例拆分 切分比例 表1占原数据的比例,范围 (0, 1) 0.8 随机数种子 正整数 12345 高级选项	选择按比例拆分，切分比例为 0.5，并设置随机数种子为 12345，便于我们重复试验结果
线性回归	字段设置　参数设置　执行调优 选择特征列 已选择 11 个字段 选择标签列 支持bigint和double类型 quality 是否稀疏格式 kv:kv格式 kv对间分隔符	选择特征列为 11 个特征字段；并选择标签列为 quality，即回归值所在的目标列
预测	字段设置　执行调优 特征列 默认全选 选择字段 原样输出列 推荐添加label列,方便评估 已选择 12 个字段 输出结果列名 prediction_result	不用设置，使用默认参数即可
回归模型评估	字段设置 原回归值 必选 quality 预测回归值 必填 prediction_score	需要设置参数“原回归值”为 quality

回归模型评估组件给出了该线性回归的两个重要指标值：均方根误差（RMSE）为 0.7524，平均绝对误差（MAE）为 0.5862。预测回归值与原回归值间的残差直方图如图 7-10 所示。

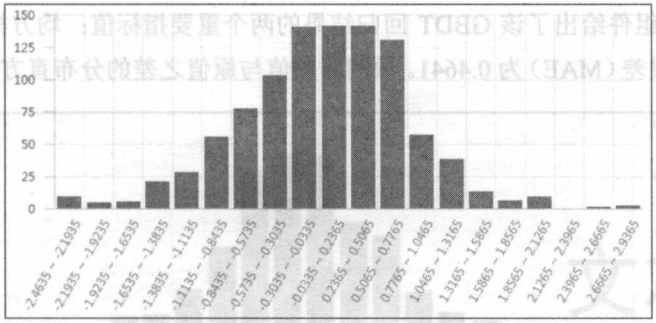


图 7-10　残差直方图

7.3　GBDT 回归

我们再尝试一个非线性的回归模型：GBDT 模型，流程与上节线性回归模型相同，只是将“线性回归”组件替换为“GBDT 回归”组件，并设置参数，在如图 7-11 所示的字段设置中，设置输入列为 11 个特征，标签列为 quality；在如图 7-12 所示的参数设置中，有两个地方需要修改，一个是损失函数类型要选择“regression loss”，另一个是叶子节点容纳的样本数，默认值 500，是针对大数据集的，我们现在考虑的数据集只有几千的规模，该参数设为 5 比较合适。

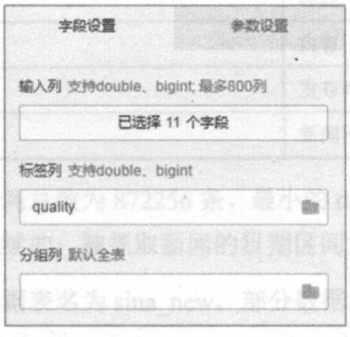


图 7-11　“GBDT 回归”组件的字段设置



图 7-12　“GBDT 回归”组件的参数设置

回归模型评估组件给出了该 GBDT 回归结果的两个重要指标值：均方根误差（RMSE）为 0.6141，平均绝对误差（MAE）为 0.4641。预测回归值与原值之差的分布直方图如图 7-13 所示。

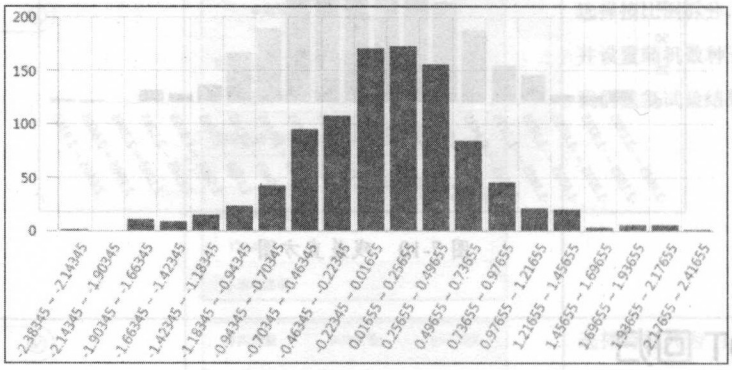


图 7-13　残差直方图

对比上一节线性回归的评估结果，无论从 RMSE、MAE 指标上，还是从回归残差的直方图上，都可以看出 GBDT 回归的效果更好。

第8章 文本分析

我们接触到的很多数据都是文本的，譬如：新闻的标题和内容、电影评论、商品描述等等。从这些文本数据中，如何寻找相似的内容？新闻中的关键词是什么？用户评论了一段话，表示的意思是喜欢还是不喜欢？从本章开始就讨论解决这样的问题。

本章会介绍一些文本分析中基本的内容，譬如：分词、词频统计、TF-IDF 等。后面的两章会基于文本内容构造特征，使用机器模型来进行赔付预测和情感分析。

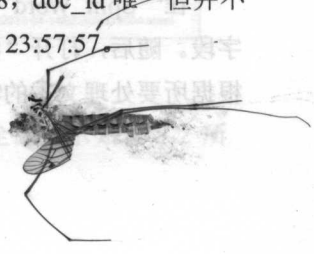
本章所用的数据是爬虫获取的新浪网页数据，包含 6 个数据列，各列的名称和含义如表 8-1 所示。

表 8-1　字段说明

字段名称	含　义
docid	爬虫获取文档时的设置的编号
category	分类
title	标题
content	内容
time	发布时间
url	新闻网址

新闻总数为 872256 条，最小的 doc_id 为 0，最大的 doc_id 为 1081628，doc_id 唯一但并不都是连续的；被抓取新闻的日期区间为 2011-01-01 00:04:30 到 2014-07-22 23:57:57。

数据表名为 sina_new，部分数据显示如图 8-1 所示。



docid	category	title	content	time	url
388280	sports	休斯顿赛伊斯内尔送砖摩纳哥将与阿尔玛德罗争冠	五号种子约翰·伊斯内尔向着美国男子红土冠军赛冠军头衔做最...	2013-04-1411:48:06	http://sports.sina.co...
388281	sports	美娜摸电影脚本回顾火箭莫雷变影帝感队翻身(图)	新浪体育讯　北京时间4月14日，现实太残酷，何不来娱乐一下...	2013-04-1414:56:57	http://sports.sina.co...
388282	video	视频：林志颖和陈若仪偶像也爱小粉丝	-	2013-04-1423:54:22	http://video.sina.co...
388283	video	意甲卡利亚里vs国米视频直播结束	-	2013-04-1420:59:32	http://video.sina.co...
388284	sports	没油、没轮胎、没积分F1中国站韦伯最想最惨No.1	新浪体育讯　排位赛燃油耗尽，正赛轮胎脱落，结局是空手而...	2013-04-1419:35:32	http://sports.sina.co...
388285	sports	澳门虎从列强沦至要保级没钱没人？比申花还难？	新浪体育讯　4月13日晚的天津奥体中心，“下课”和“解散”声在...	2013-04-1401:09:15	http://sports.sina.co...
388286	sports	意甲-佩罗伦萨两替补发威客场2-0保级战提前纳3-2	新浪体育讯　北京时间4月14日2时45分(意大利当地时间13日2...	2013-04-1404:43:35	http://sports.sina.co...
388287	news	32岁女子3年考4次公务员怀孕时也复习到深夜	渴望“体制内幸福生活”怀孕时也复习到深夜　3年“考碗”路：...	2013-04-1407:59:54	http://news.sina.co...
388288	slide.ent	C罗女友超模伊莲娜比基尼热辣抖腰(图)	-	2013-04-1410:19:04	http://slide.ent.sina....
388289	news	伊拉克2名省议会选举候选人遭袭丧生	新华网巴格达4月14日电(记者张淑惠梁有昶)伊拉克警方14日说...	2013-04-1420:06:43	http://news.sina.co...
388290	sports	联盟篇一防守队？快船不怕季后赛碰到就照这样打	新浪体育讯　91比87，快船(微博)客场捕鲸，拿下了这场事关季...	2013-04-1411:11:10	http://sports.sina.co...
388291	finance	凤凰古城收费风波调查：148元门票33元归政府	“凤凰古城收取148元门票致商铺关门抗议，收费几天央视波不...	2013-04-1404:36:47	http://finance.sina.c...
388292	video	视频：文章遭女主持调戏天然呆萌翻众人	-	2013-04-1409:34:59	http://video.sina.co...
388293	video	进球视频-恒大反击8024模联埃尔克莱抽空门得手	-	2013-04-1416:58:18	http://video.sina.co...
388294	video	视频：鸟叔PSY新单《GENTLEMAN》MV首播	-	2013-04-1416:25:01	http://video.sina.co...
388295	news	何足道：出租车调价能否从高峰时段开始	■来论　在节假日、早晚高峰等情况下，有乘客可能会愿意多...	2013-04-1402:39:47	http://news.sina.co...
388296	sports	南京大学获全国大学生五人制足球联赛江苏赛区冠...	新华社南京4月14日体育专电（记者王恒志）特步中国大学生五...	2013-04-1419:19:13	http://sports.sina.co...

图 8-1　数据表 sina_new 的内容

8.1　分词

基于 AliWS（Alibaba Word Segmenter 的简称）词法分析系统，对指定列对应的文章内容进行分词，分词后的各个词语间以空格作为分隔符，若用户指定了词性标注或语义标注相关参数，则会将分词结果、词性标注结果和语义标注结果一同输出，其中词性标注分隔符为“/”，语义标注分隔符为“|”。目前 AliWS 仅支持中文淘宝分词和互联网分词。

选择“Split Word”组件，如图 8-2 所示配置工作流，需要注意的是，原始文本接入左边的输入桩；右边的输入桩预留给自定义词典，词典定义为一列的数据表，每一行是一个词。

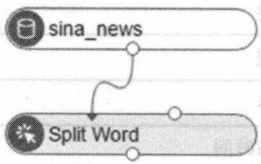


图 8-2　分词组件的连接

在“Split Word”组件的“字段设置”选项卡中，选择需要进行分词的 title 字段和 content 字段。随后，打开“参数设置”选项卡，里面有很多参数可以选择，如图 8-3 所示，用户可以根据所要处理文本的特点，灵活选择配置相应参数。

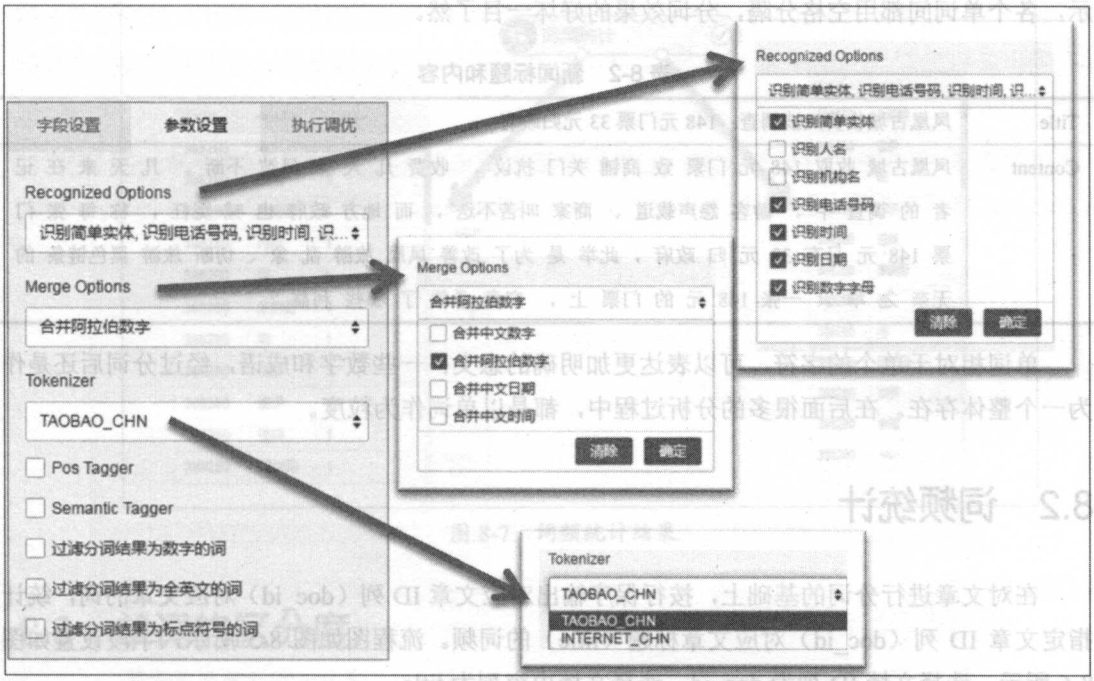


图 8-3 “Split Word”组件的参数设置

分词组件的结果，如图 8-4 所示，选中的字符串字段用分词后的形式替换。

docid	category	title	content	time	url
388280	sports	伏斯顿 囊伊斯 内尔 逆转 摩纳哥 将 与 ...	五 号 种子 约翰 · 伊斯 内尔 ...	2013-04-1411:48:06	http://sports.sina.com.cn/t/2013-04-14/11486518329.shtml
388281	sports	美媒 揭 电影 脚本 回顾 火箭 翼雷 变 影...	新浪 体育 讯　北京 时间 4...	2013-04-1414:56:57	http://sports.sina.com.cn/nba/2013-04-14/14566518550.shtml
388282	video	视频 ：林志颖 和 陈若仪 偶像 也 要 小 ...	-	2013-04-1423:54:22	http://video.sina.com.cn/p/ent/v/m/2013-04-14/23546230096...
388283	video	意甲 卡利亚里 vs 国米 视频 直播 结束	-	2013-04-1420:59:32	http://video.sina.com.cn/p/sports/z/v/2013-04-14/205962300...
388284	sports	没 油、没 轮胎、没 积分 F1 中国 站 ...	新浪 体育 讯　排位赛 燃油...	2013-04-1419:35:32	http://sports.sina.com.cn/f1/2013-04-14/19356518984.shtml
388285	sports	澳门 虎 从 列强 沦 至 要 保级 没 钱 没 ...	新浪 体育 讯　4月 13日 晚...	2013-04-1401:09:15	http://sports.sina.com.cn/j/2013-04-14/01096517566.shtml
388286	sports	意甲 - 佩罗 伦萨 两 替 补发 威 客场 2 - 0...	新浪 体育 讯　北京 时间 4...	2013-04-1404:43:35	http://sports.sina.com.cn/g/seriea/2013-04-14/04436517800...
388287	news	32 岁 女子 3 年 考 4 次 公务员 怀孕 时 ...	渴望 “ 体制 内 幸福 生活 ”怀...	2013-04-1407:59:54	http://news.sina.com.cn/s/2013-04-14/075926820765.shtml
388288	slide.ent	C 罗 女友 超模 伊莲娜 比基尼 热舞 抖 ...	-	2013-04-1410:19:04	http://slide.ent.sina.com.cn/star/slide_4_704_55660.html
388289	news	伊拉克 2 名 省 议会 选举 候选人 遭 袭 ...	新华网 巴格达 4月 14日 电 (...	2013-04-1420:06:43	http://news.sina.com.cn/w/2013-04-14/200626824151.shtml
388290	sports	联盟 第一 防守 队 ？ 快船 不怕 季后赛 ...	新浪 体育 讯　91 比 87 ，...	2013-04-1411:11:10	http://sports.sina.com.cn/nba/2013-04-14/11116518307.shtml
388291	finance	凤凰 古城 收费 风波 调查 ：148 元 门票 ...	“ 凤凰 古城 收取 148 元 门票...	2013-04-1404:36:47	http://finance.sina.com.cn/china/20130414/043615140107.s...
388292	video	视频 ：文章 遭 女 主持 调戏 天然 呆 萌 ...	-	2013-04-1409:34:59	http://video.sina.com.cn/p/ent/m/c/2013-04-14/09346229954...
388293	video	进球 视频 - 恒大 反击 8024 横 敲 埃尔 ...	-	2013-04-1416:58:18	http://video.sina.com.cn/p/sports/j/v/2013-04-14/165862299...
388294	video	视频 ：鸟 叔 PSY 新 单 《 GENTLEM...	-	2013-04-1416:25:01	http://video.sina.com.cn/p/ent/y/2013-04-14/162562299751...
388295	news	何 足道 ：出租车 调价 能否 从 高峰 时 ...	■ 来 论　在 节假日 、早 ...	2013-04-1402:39:47	http://news.sina.com.cn/pl/2013-04-14/023926819094.shtml

图 8-4 分词组件运行结果

为了看清分词后的效果，我们选取其中一条新闻，并将其标题和内容全部显示如表 8-2 所

示，各个单词间都用空格分隔，分词效果的好坏一目了然。

表 8-2 新闻标题和内容

Title	凤凰古城收费风波调查：148 元门票 33 元归政府
Content	凤凰古城收取 148 元门票致商铺关门抗议，收费几天来风波不断。几天来在记者的调查中，游客怨声载道、商家叫苦不迭，而地方政府也喊冤枉，称每张门票 148 元只有 33 元归政府，此举是为了改善凤凰旅游乱象、切断旅游黑色链条的无奈之举。一张 148 元的门票上，究竟承载了哪些利益？

单词相对于单个的字符，可以表达更加明确的意义；一些数字和成语，经过分词后还是作为一个整体存在。在后面很多的分析过程中，都是以单词作为粒度。

8.2 词频统计

在对文章进行分词的基础上，按行保序输出对应文章 ID 列（doc_id）对应文章的词，统计指定文章 ID 列（doc_id）对应文章标题（title）的词频。流程图如图 8-5 所示，字段设置如图 8-6 所示，选择文档 ID 列为 doc_id，选择文档内容列为 title。

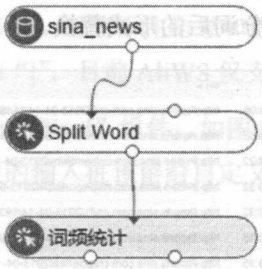


图 8-5 “词频统计”组件的连接

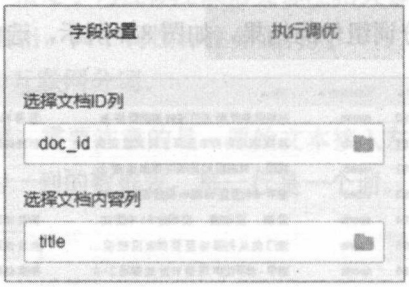


图 8-6 “词频统计”组件的字段设置

“词频统计”组件的结果如图 8-7 所示。每个单词前都有其对应的 doc_id，我们可以知道该单词是由哪篇文章计算来的，左边输出每篇文章标题出现的所有单词及其出现的个数，注意单词之间是无序的；右边输出有序的概念，每个单词占一行，按照文章标题中出现的次序，依次输出，最后的字符串“</s>”是句子的结束符。

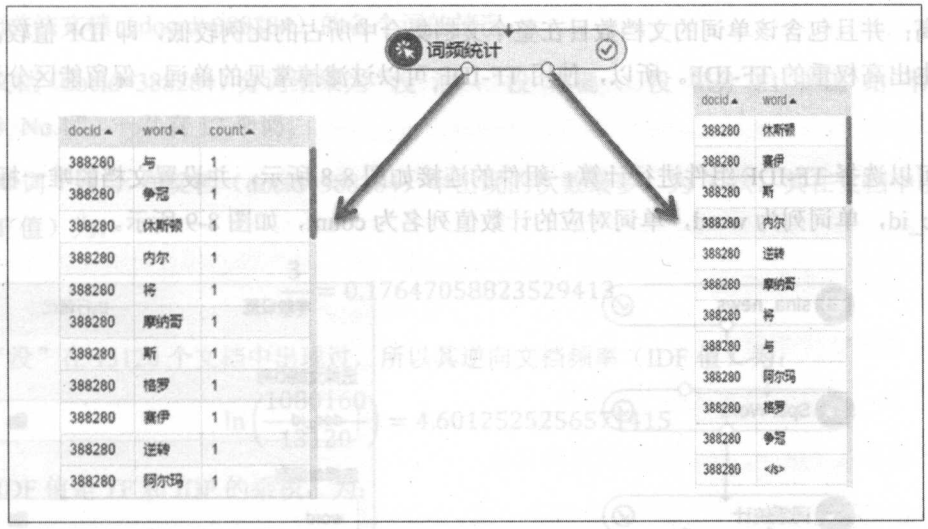


图 8-7　词频统计结果

8.3　单词的区分度

我们先介绍几个重要概念：

1）TF-IDF（Term Frequency - Inverse Document Frequency）是通过计算单词在文档中的频率，进而得到单词的权重，可以用来评估单词对于一个文档集或一个语料库其中一份文档的重要程度。单词的重要性与它在文档中出现的次数成正比，但同时会与它在语料库中出现的频率成反比。

2）TF（Term Frequency），即单词在该文档中出现的频率，公式为

$$\frac{\text{单词在该文档出现的次数}}{\text{该文档中单词的总数}}$$

显然，TF 值越大表示这个单词对该文档越重要。

3）IDF（Inverse Document Frequency）是逆向文档频率，一个文档库中，一个分词出现在的文档数越少越能和其他文档区别开来。算法是

$$\ln\left(\frac{\text{总文档数}}{\text{出现该单词的文档数}}\right)$$

TF-IDF 值的计算就是将 TF 和 IDF 的值相乘。对于某一特定文档内的高频单词，即 TF 的

值较高；并且包含该单词的文档数目在整个文档集合中所占的比例较低，即 IDF 值较高；则可以产生出高权重的 TF-IDF。所以，使用 TF-IDF 可以过滤掉常见的单词，保留能区分该文档的单词。

可以选择 TF-IDF 组件进行计算，组件的连接如图 8-8 所示，并设置文档的唯一标识 ID 列为 doc_id，单词列为 word，单词对应的计数值列名为 count，如图 8-9 所示。

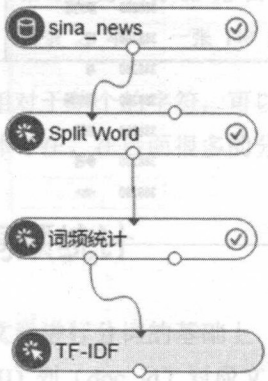


图 8-8　TF-IDF 组件的连接

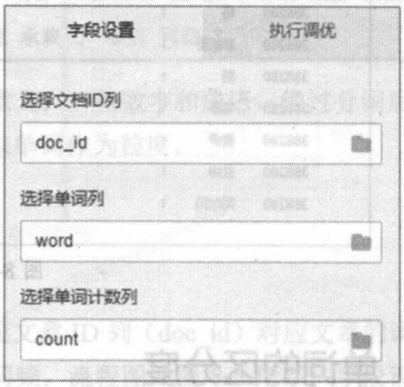


图 8-9　TF-IDF 组件的字段设置

计算结果如图 8-10 所示，其中部分数据列的含义为：数据列“count”为单词在该文档中出现的次数；“total_word_count”为该文档中的单词总数；“doc_count”为含有该单词的文档的总数；“total_doc_count”为全部文档的数量。

docid	word	count	total_word_count	doc_count	total_doc_count	tf	idf	tfidf
388283	结果	1	7	2758	1080160	0.14285714285714285	5.970358677922451	0.8529083825603501
388283	视频	1	7	180560	1080160	0.14285714285714285	1.7888013247842802	0.2555430463977543
388284	F1	1	17	1495	1080160	0.058823529411764705	6.582738250450911	0.3872198708534765
388284	No.1	1	17	263	1080160	0.058823529411764705	8.320465704097932	0.4894391906458424
388284	,	2	17	526	1080160	0.11764705882352941	7.627318523537988	0.8973315910044691
388284	中国	1	17	50534	1080160	0.058823529411764705	3.0622180802282832	0.18013047530754606
388284	慧	1	17	456	1080160	0.058823529411764705	7.770126926761312	0.45706628980948893
388284	惨	1	17	477	1080160	0.058823529411764705	7.7251032453873565	0.4544178379639621
388284	最	2	17	10844	1080160	0.11764705882352941	4.6012525256571415	0.541323826547899
388284	没	3	17	13120	1080160	0.17647058823529413	4.410726673777617	0.7783635306666383
388284	油	1	17	1235	1080160	0.058823529411764705	6.77379348721362	0.3984584404243306
388284	积分	1	17	455	1080160	0.058823529411764705	7.772322317324747	0.4571954304308674 7
388284	站	1	17	4281	1080160	0.058823529411764705	5.5306778301307515	0.32533399000769125
388284	链睛	1	17	1	1080160	0.058823529411764705	13.892619736275698	0.8172129256632763
388284	韦伯	1	17	221	1080160	0.058823529411764705	8.494457034757945	0.4996739432210555 7
388285	人	1	18	39091	1080160	0.05555555555555555	3.3189721958256317	0.18438734421253508

图 8-10　TF-IDF 组件的计算结果

我们关注文档（docid=388284）的各个词的情况：

1）文档（docid=388284）分词结果为“没 油 、 没 轱辘 、 没 积分 F1 中国 站 韦伯 最 悲 最 惨 No.1”，一共有 17 个词。

2）单词“没”，在文档（docid=388284）中出现的次数最多，为 3 次。其在文档中出现的频率（TF 值）为：

$$\frac{3}{17} = 0.17647058823529413$$

但“没”在 13120 个文档中出现过，所以其逆向文档频率（IDF 值）为：

$$\ln\left(\frac{1080160}{13120}\right) = 4.6012525256571415$$

TF-IDF 值是 TF 和 IDF 的乘积，为：

$$0.17647058823529413 \times 4.6012525256571415 = 0.7783635306666383$$

3）单词“轱辘”，在文档（docid=388284）中出现的次数只有一次，其在文档中出现的频率（TF 值）会较小，为：

$$\frac{1}{17} = 0.058823529411764705$$

由于“轱辘”只在 1 个文档中出现过，所以其逆向文档频率（IDF 值）较大，为：

$$\ln\left(\frac{1080160}{1}\right) = 13.892619736275698$$

TF-IDF 值是 TF 和 IDF 的乘积，为：

$$0.058823529411764705 \times 13.892619736275698 = 0.8172129256632763$$

4）汇总一下，单词“轱辘”和“没”在文档（docid=388284）里 TF-IDF 值最高。单词“轱辘”是由于其在其他文档中没有出现，逆向文档频率（IDF 值）突出，从而使 TF-IDF 值很大；单词“没”是由于其在文档中重复出现 3 次，即，其在文档中出现的频率（TF 值）显著，从而使 TF-IDF 值也很大。

8.4 字符串比较

字符串不能像数字那样定义大小，但是我们可以看出哪些字符串更接近，这种“接近”是通过字符串间的距离和相似度来衡量的，常用的有编辑距离（Levenshtein Disrance）、最长公共

子串（Longest Common Substring，缩写为 LCS）距离、余弦相似度、Jaccard 相似度等。

本节会使用编辑距离（Levenshtein Disrance）来实验字符串比较的操作。编辑距离，是由俄国科学家 Levenshtein 提出的，故又叫 Levenshtein 距离，指的是在两个字符串之间，由一个字符串转换成另一个字符串所需要的最少的编辑操作次数。

编辑操作分为以下三种：

1）插入一个字符。

2）删除一个字符。

3）将一个字符替换成另一个字符。

下面举几个例子，加深对此距离的理解。

- ab 和 ab 需要有 0 个操作，字符串一样。
- ab 和 a 需要有 1 个删除操作。
- a 和 ab 需要有 1 个插入操作。
- ab 和 ac 需要有 1 个替换操作。

编辑距离相似度的公式如下，单位 1 减去编辑距离与最大字符串长度的比值。

$$\text{Simi}_{\text{Edit}} = 1 - \frac{EditDistance}{\text{Max}(Length1, Length2)}$$

我们通过例子来实验两种常见的操作：对于给定的字符串对，计算其编辑距离；在字符串的参考数据之中，计算与目标字符串最“相似”的 N 个字符串。分别需要使用“字符串相似度”组件、“字符串相似度-topN”组件，整体的流程图如图 8-11 所示，这个实验有两个分支。左边的分支比较简单，只有一个节点；右边的分支由三个节点构成。

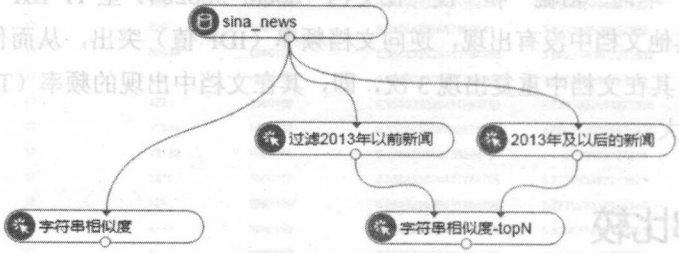


图 8-11　字符串比较的整体流程

左边的分支只有一个组件 字符串相似度，比较同一记录中的两个不同字符串字段的内容，我们希望看一下，新闻标题与新闻内容间的编辑距离是多少，这需要对组件进行设置。在“字段设置”中，如图 8-12 所示，选择相似度计算中第一列的列名为“title”，第二列的列名为“content”，使用默认的输出表中相似度的列名“output”。需要注意的是，完成这些设置，结果表中只输出一列“output”，即默认不输出参加计算的两个字符串列，如果我们希望看到它们，需要在输出表追加的列名中选择“title”和“content”。在“参数设置”中，如图 8-13 所示，选择相似度计算方法为“levenshtein”，即编辑距离。



图 8-12 字符串相似度组件的字段设置

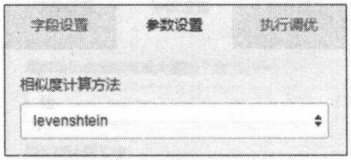


图 8-13 字符串相似度组件的参数设置

运行的结果，如图 8-14 所示，第一列为新闻标题，第二列为新闻内容，第三列是标题与内容间的编辑距离。

title	content	output
女排大奖赛澳门站12人名单确定朱婷将担起重任	新浪体育讯 北京时间7月29日，郎平(微博)率领的中国女排国家队在国家体育总局训练局排球馆举行了媒体公开...	1334.0
《一夜惊喜》倪虹洁扮丑上演丑女逆袭	新浪娱乐讯昨日(7月29日)，由寰亚电影(微博)、华夏视听、鹊影画联合出品，导演金依萌原创剧本并执导的电影...	834.0
易建联：困难再多也要卫冕伤停乐观一周内恢复	本报讯昨天下午5点，中国男篮从天坛公寓出发，启程赶往菲律宾的马尼拉，这次亚锦赛之旅对中国男篮来说困难...	663.0
澳大利亚外长：中国复兴对澳是极好机会	原标题：澳大利亚外长指出中国复兴对澳大利亚是极好机会 国际在线报道(记者徐位毅)澳大利亚外长卡尔29日...	681.0
央行政策会议前欧股高开	新浪财经讯北京时间7月29日下午消息，周一欧股高开，本周美国、欧洲、英国央行将举行货币政策会议。 北...	437.0
郑州富士康选工会组长员工拿白条茫然投票	有分析指出，富士康推动工会选举的目的是应付采购商的审查及降低管理成本，而非改善工人权益状况 本报...	1345.0
梅新育：马丁雷，阅论炮制的种族歧视	美国佛罗里达州费福德市社区义务警卫乔治·齐默尔曼枪杀黑人青年特雷弗·马丁案在美国社会引起了轩然大波。由...	1603.0
刘戈：适应负面新闻，保持整体理性	刘戈 随着事实上的言论开放，作为世界上人口最多的国家，中国人需要适应这样一种局面：从新闻上看这是...	1329.0
约翰逊拜访女友父亲餐厅小两口随时秀恩爱(多图)	新浪体育讯 北京时间7月29日消息，达斯汀·约翰逊和女友梅林娜在多伦多的一家餐厅前合影。而这家餐厅正好...	331.0
新京报：共同维护法律底线，向暴徒坚决说不	■社论 个人追求的正义，应该以守法为底线，而不是“以武犯禁”、“绿林好汉式的正义”，而社会也应为公民的...	1242.0
布兰妮晒与爱子合影孩子颜质可爱(图)	新浪娱乐讯“小甜甜”布兰妮(微博)于7月29日在微博晒出了自己与两个儿子的合影，称要带他们去参加《蓝精灵2》...	111.0
卡塔尔天价外援加盟3周即去世心脏病突发年仅27	新浪体育讯 北京时间7月29日晚，在7月初刚刚以1000万英镑的身价从墨西哥美洲队加盟卡塔尔阿尔萨什队的外...	699.0

图 8-14 字符串相似度组件的计算结果

选择了两条字数较少的新闻，如表 8-3 所示，感兴趣的读者可以验证一下其编辑距离。注意每条新闻内容的开头都有两个空格，也需要计算到字符数。

表 8-3 两个编辑距离的例子

title	content	output
一季度城镇居民人均可支配收入 6796 元	一季度城镇居民人均可支配收入 6796 元，同比名义增长 14.0%。-国家统计局	21.0
永泰地产停牌以待公布影响股价资料	新浪财经讯 4 月 13 日消息，永泰地产公布，今日午后停牌，以待公布有关可能影响股份价格资料的公布。	33.0

右边的分支是要将两个字符串集合进行比较，这里是将新闻按其报道的时间，分为 2013 年前的新闻和 2013 年及以后的新闻，可以通过两个“过滤与映射”组件实现，具体设置如图 8-15 和图 8-16 所示，主要是使用了日期比较函数 datediff()，将时间分割点“2013-01-01 00:00:00”与记录中的新闻报道时间字段“time”相比较。

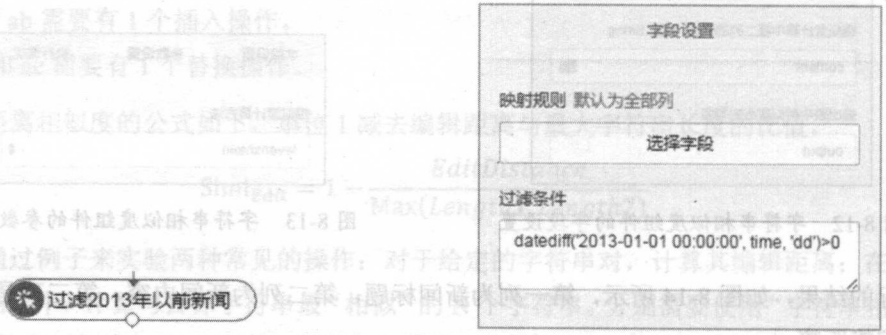


图 8-15 “过滤 2013 年以前新闻”组件的参数设置

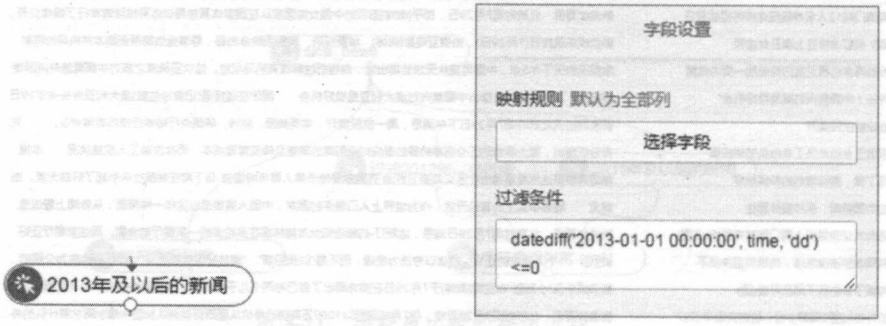
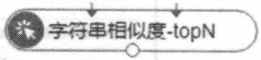


图 8-16 “2013 年及以后的新闻”组件的参数设置

我们是对新闻标题进行比较，对于 2013 前的每条新闻，可以找出 N 条与其最“接近”的 2013 年及以后的新闻。使用“字符串相似度-topN”组件，

的具体参数设置：在“字段设置”中，选择相似度计算中左表的列名为“title”，映射表的列名为“title”；使用默认的输出表中相似度的列名“output”，如图 8-17 所示。在“参数设置”中，选择相似度计算方法为“levenshtein_sim”，即编辑距离相似度，如图 8-18 所示。

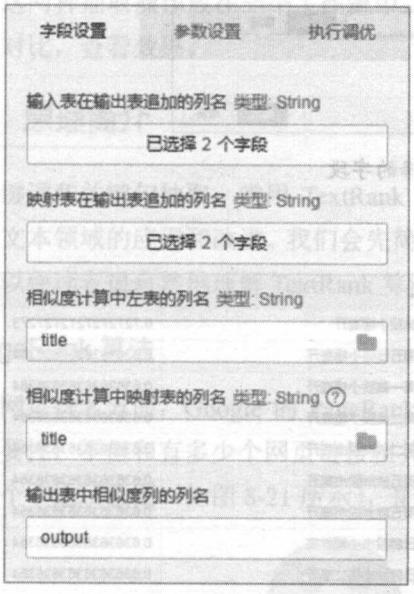


图 8-17 组件字段设置

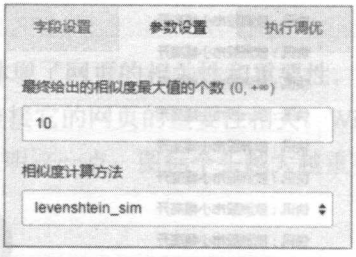


图 8-18 组件参数设置

需要注意的是，完成这些设置，结果表中只输出一列“output”，即默认不输出参加计算的两个字符串列，如果我们希望看到它们及对应的 doc_id，需要在输出表追加的列名中选择上“doc_id”和“title”，但由于输入表和映射表都需要输出这两列，列名便会重复，解决此冲突的办法是定义它们各自在输出表中的新列名，在选列窗体上有此功能，如图 8-19 所示，右侧显示的输出字段可以重命名。

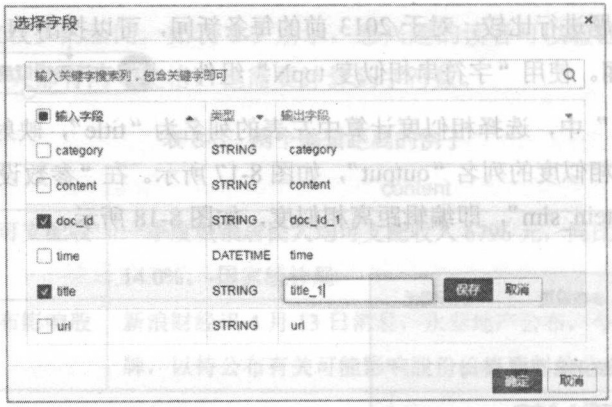


图 8-19　重命名选择的字段

设置好各组件后，运行结果如图 8-20 所示。

doc_id_1	title_1	doc_id_2	title_2	output
10416	快讯：欧洲股市小幅高开	454671	快讯：美股小幅高开	0.7272727272727273
10416	快讯：欧洲股市小幅高开	148766	快讯：周五日经小幅高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	177471	快讯：周一美股小幅高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	670882	快讯：周二美股小幅高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	754818	快讯：周二欧洲股市高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	467484	快讯：周五欧洲股市高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	489965	快讯：周五欧洲股市高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	494075	快讯：日韩股市小幅收高	0.6363636363636364
10416	快讯：欧洲股市小幅高开	495315	快讯：日韩股市周二高开	0.6363636363636364
10416	快讯：欧洲股市小幅高开	813471	快讯：周五美股小幅高开	0.6363636363636364
10417	郑晓龙后甄嬛时期两部现代戏均热卖	529611	郑晓龙否认甄嬛拍续集儿子在美学特效	0.2941176470588235
10417	郑晓龙后甄嬛时期两部现代戏均热卖	805827	郑晓龙谈甄嬛传被正名：有争议即成功	0.2941176470588235
10417	郑晓龙后甄嬛时期两部现代戏均热卖	711203	郑晓龙否认拍甄嬛续集希望孙俪合作新戏	0.2777777777777778
10417	郑晓龙后甄嬛时期两部现代戏均热卖	859537	郑晓龙：甄嬛传应再提价让烂片彻底死掉	0.2777777777777778
10417	郑晓龙后甄嬛时期两部现代戏均热卖	469364	郑晓龙：脱离时代批甄嬛不讲理很恶心	0.23529411764705888

图 8-20　相似度 TopN 组件的输出结果

可以看到，“快讯：欧洲股市小幅高开”与“快讯：美股小幅高开”的编辑距离相似度最大为 0.72727272727273，这两个标题间的编辑距离为 3，也是最小的。新闻标题“郑晓龙后甄嬛时期两部现代戏均热卖”与后面相似的 5 个标题，共同出现的词都只有“郑晓龙”和“甄嬛”，但由于整个标题长度的差异，即这两个词间文字个数的不同造成了编辑距离相似度上的差异。

8.5 抽取关键词、关键句

文档中的关键词和关键句，是对详细文档内容的一种提炼和总结，可以体现文档的基本内容，帮助用户迅速了解文档所关注的领域，所持的基本观点。

本小节会从常用的关键内容计算原理入手，帮助用户更深刻地理解“关键”的含义；然后，还是将这两种抽取操作放在一个工作流中，可以分别看到各操作的结果，并可以针对原文档的内容做对比，查看效果。

8.5.1 原理简介

关键词和关键句抽取，常用 TextRank 算法，该算法可以看作是著名的网页排名 PageRank 算法在文本领域的应用和改进。我们会先简要地介绍 PageRank 算法，然后再把文本的特点融入其中，以便读者很自然地理解 TextRank 算法。

1. PageRank 算法

在网页排名方面，Google 的 PageRank 算法很好地体现了网页的相关性和重要性，一个网页的重要性，不但和有多少个网页链接相关，还与每个链接它的网页的重要性相关。Wikipedia 上的一个卡通示意图（如图 8-21 所示），能比较形象地说明这一点，图标个头越大越重要。

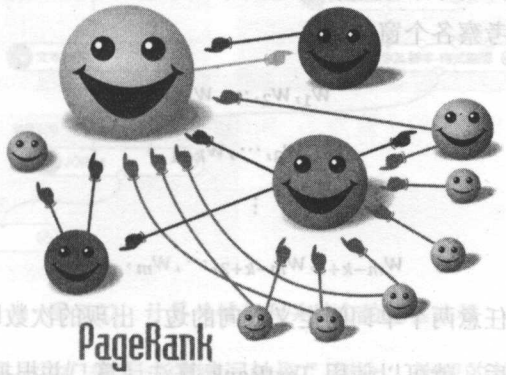


图 8-21 PageRank 卡通演示图

图片来源：http://en.wikipedia.org/wiki/PageRank#mediaviewer/File:PageRank-hi-res.png。

PageRank 的计算公式如下：

$$\mathrm{PR}(P_i) = (1-d) + d \cdot \sum_{P_k \in In(P_i)} \frac{1}{|\mathrm{Out}(P_k)|} \mathrm{PR}(P_k)$$

这里$\mathrm{PR}(P_i)$是网页P_i的 PageRank 值；d 是阻尼系数，一般设置为 0.85；$\mathrm{In}(P_i)$是存在指向网页P_k的链接的网页集合；$Out(P_k)$是网页P_k中的链接存在的链接指向的网页的集合。

2. TextRank 算法

TextRank 算法是基于 PageRank 算法的，可以看作是 PageRank 在自然语言处理领域的应用，主要用于为文本生成关键字和摘要。

正规的 TextRank 公式在 PageRank 的公式的基础上，引入了边的权值的概念，代表两个单词或句子的相似度，公式如下：

$$\mathrm{TR}(T_i) = (1-d) + d \cdot \sum_{T_k \in \mathrm{In}(T_i)} \frac{w_{ki}}{\sum_{T_j \in \mathrm{Out}(T_k)} w_{kj}} \mathrm{TR}(T_k)$$

这里$\mathrm{TR}(P_i)$是单词或句子T_i的 TextRank 值；d 是阻尼系数，一般设置为 0.85；$\mathrm{In}(T_i)$是指向T_i的单词或句子集合；$\mathrm{Out}(T_k)$是被T_k指向的单词或句子集合。

1）计算关键词。

首先要将文档进行分词，并进行停用词过滤等操作。单词之间的联系是通过它们是否在句子中相邻出现来定义的，具体做法是，定义一个窗口长度k，对于每个句子的单词$w_1, w_2, w_3, \cdots, w_{m-1}, w_m$，考察各个窗口

$$w_1, w_2, \cdots, w_k$$

$$w_2, w_3, \cdots, w_{k+1}$$

$$\vdots$$

$$w_{m-k+1}, w_{m-k+2}, \cdots, w_m,$$

为每个窗口中出现的任意两个单词间定义双向的边，出现的次数即为其权重。

在定义了单词和联系后，就可以使用 TextRank 算法计算，并根据 rank 值确定单词的重要性。

2）计算关键句。

关注的对象是文档拆解出的句子，并认为所有句子间都有关联，将句子间的相似度作为其权值。

常用的句子间相似度有以下几种。

- 最长公共子串（Longest Common Subsequence）相似度。
- 编辑距离（Levenshtein Distance）相似度。
- 余弦（cosine）相似度。
- 谱核相似度。

8.5.2　完整流程

我们还是使用新浪新闻数据集，针对每条新闻内容，计算出关键词和关键句，并将计算结果与原始内容放在一起进行比较、分析。完整流程如图 8-22 所示，包括三个部分：关键词抽取、关键句抽取、结果汇总。

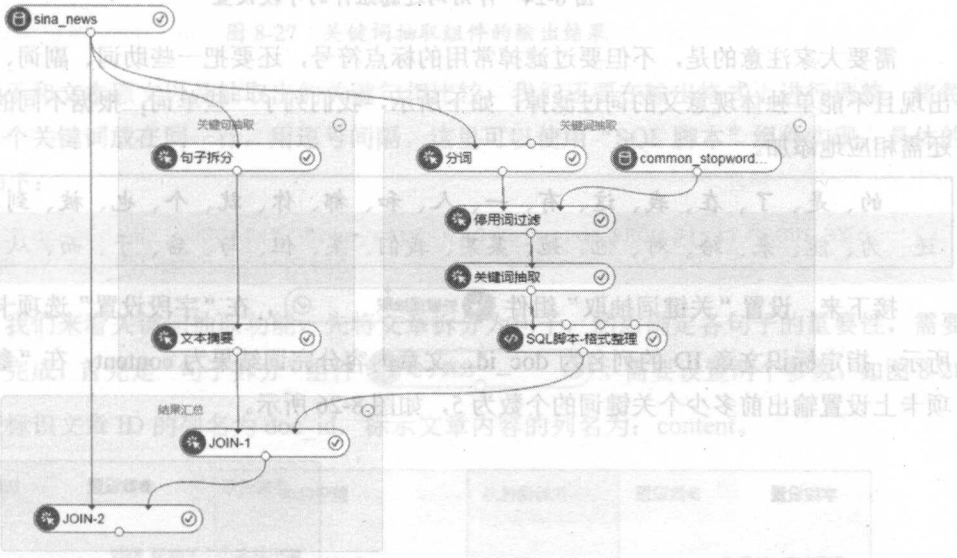


图 8-22　计算关键词和关键句的完整流程

按照算法的介绍顺序，我们先看“关键词抽取”部分。处理的粒度单位是单词，所以先要使用分词组件，随后使用我们熟悉的“停用词过滤”组件。参数设置如图 8-23 和图 8-24 所示，需要分词的字段和过滤停用词的字段都设置为 content，即新闻内容列。

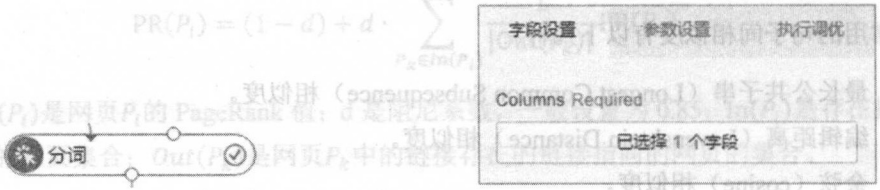


图 8-23　分词组件的字段设置

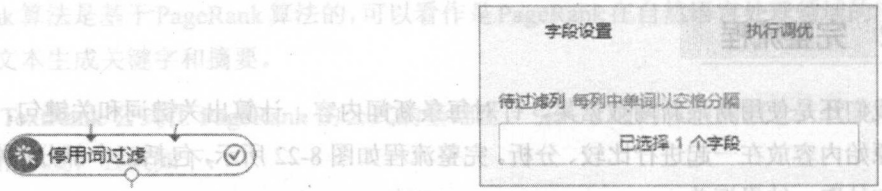


图 8-24　停用词过滤组件的字段设置

需要大家注意的是，不但要过滤掉常用的标点符号，还要把一些助词、副词、代词等经常出现且不能单独体现意义的词过滤掉，如下所示，我们列了一些单词，根据不同的场景，用户还需相应地添加。

的、是、了、在、我、这、有、一、人、和、都、你、就、个、也、被、到、要、上、还、为、能、来、给、对、他、她、某某、我们、某、但、与、后、于、而、从、其、

接下来，设置“关键词抽取”组件 

，在“字段设置”选项卡，如图 8-25 所示，指定标识文章 ID 的列名为 doc_id，文章内容分完词结果为 content；在“参数设置”选项卡上设置输出前多少个关键词的个数为 5，如图 8-26 所示。

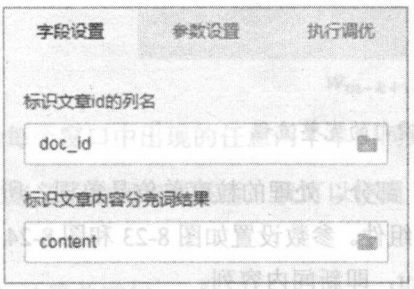


图 8-25　关键词抽取组件的字段设置

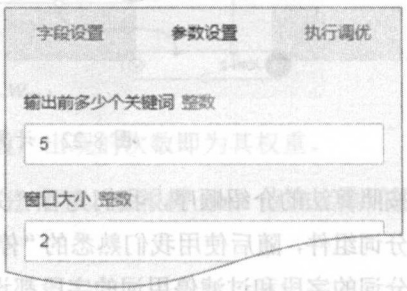


图 8-26　关键词抽取组件的参数设置

此组件的输出结果如图 8-27 所示，对于每个 doc_id 输出 5 个关键词，每个关键词还对应着一个权重值。

doc_id ▲	keywords ▲	weight ▲
1006006	工作	0.0391992432329494 9
1006006	监察	0.02682474438726408
1006006	特邀	0.023050841453838778
1006006	监察部	0.019391851619912095
1006006	监察员	0.013589089708689564
1006019	皇马	0.062238462322817614
1006019	迭戈	0.02154795059570862
1006019	洛佩斯	0.02051314022111912

图 8-27　关键词抽取组件的输出结果

为了便于和文本原文以及抽取出的关键句相比较，我们还要在输出格式上进行调整，将每个文本的 5 个关键词放在同一行，用逗号间隔。这里可以使用“SQL 脚本”组件实现，具体的 SQL 脚本如下：

```
SELECT doc_id, WM_CONCAT(", ",keywords) AS keywords FROM ${t1} GROUP BY doc_id
ORDER BY doc_id DESC LIMIT 100
```

下面，我们来看关键句抽取功能，先将文章拆分为句子，然后确定各句子的重要性，需要分两个组件完成，首先是“句子拆分”组件 

，需要设置两个参数，如图 8-28 所示，设置标识文章 ID 的列名为 doc_id，标示文章内容的列名为：content。

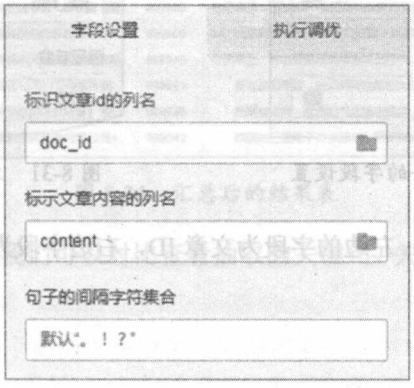


图 8-28　句子拆分组件的字段设置

句子拆分结果如图 8-29 所示，包含 doc_id 和 sentence 两个字段，分别记录文章 ID 和句子内容。

doc_id ▲	sentence ▲
135289	新浪体育讯　北京时间7月29日，郎平[微博]率领的中国女排国家队在国家体育总局训练局排球馆举行了媒体公开课。
135289	30日，郎平将率领中国女排部分队员赴澳门参加大奖赛首周的争夺。
135289	大奖赛首站新人担纲　明天中国女排部分队员将出征在澳门举行的世界女排大奖赛首站比赛，今天的媒体公开训练也是...
135289	全运会结束后国手们分批归队，最晚归队的是天津、辽宁、上海和江苏的国家队队员，连续征战有没有时间调休让姑娘们个个...
135289	上周练的时间不长，主要是在身体和力量上做储备，这周感觉好一些，但明天就要出发了。
135289	"　澳门站的比赛郎平确定的12人名单中没有队长惠若琪、主攻殷娜[微博]等人，而首次入选国家队的小将朱婷[微博]、乔婷和...
135289	谈到大奖赛的人员配备，郎平说："参加澳门站比赛的队员打两站，下一组打后两站，当然现在还不知道有没有第四周的比赛。
135289	"对于世界女排大奖赛，郎平的意图是给新人更多参赛的机会，而没有过多考虑能否进入大奖赛总决赛。

图 8-29　句子拆分组件的运行结果

然后设置“文本摘要”组件 

，在“字段设置”栏，指定标识文章 ID 的列名为 doc_id，句子列为 sentence，即前一个组件的输出列，如图 8-30 所示；在“参数设置”栏，设置输出前多少个关键词的个数为 1，并选择句子相似度的计算方法为 lcs_sim，即最大公共子串相似度，如图 8-31 所示。

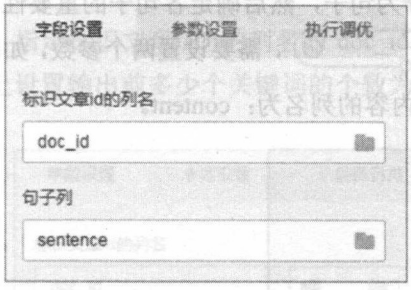


图 8-30　文本摘要组件的字段设置

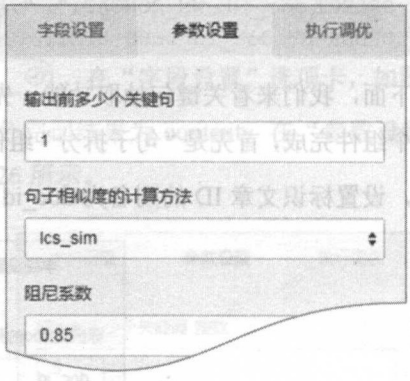


图 8-31　文本摘要组件的参数设置

计算结果如图 8-32 所示，左边的字段为文章 ID，右边字段为选择的关键句。

doc_id	abstract
100001	片方精心挑选的"去爱我"上映日期，更是传递出勇敢追爱的新时代爱情态度。
1000018	"内援的引进比较难，主力队员谁都不会放，为数不多的几个优秀内援各家俱乐部都在争抢，报价更是堪比外援。
1000020	2013年六大围棋世界大赛收官，中国前所未有包揽全部冠军，韩国围棋大溃败。
1000031	争取到2017年底，潍坊市金融业增加值占生产总值比重达到8%以上　■本报记者赵宁　近日，山东省潍坊市委...
1000047	律师王磊称，他去郑州市食品药品监督管理局高新分局查资料，被一官员咬伤。
1000055	英德警方称，当时围观的人员不止100多人，应当有几百人，但真正参与斗殴的人只有十多人。
1000080	问起春晚的消息，马星连连抱歉表示，今年春晚自己并没有太多过问，也没有听说什么传闻。
1000090	"他指出，今后辽宁舰要时常靠泊这一军港，已基本形成赛路，可出海训练了。

图 8-32　文本摘要组件的计算结果

最后，为了便于比较结果，我们将原始文章标题、内容，与计算抽取的关键词、关键句，根据文章 ID 连接起来。可以使用两个“JOIN”组件实现，如图 8-33 所示。

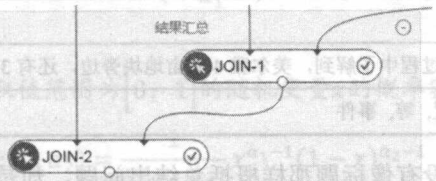


图 8-33　结果汇总流程图

汇总后的结果表，如图 8-34 所示，各字段依次为标题、内容、文章 ID、关键句、关键词。

title	content	doc_id	abstract	keywords
湖北4个调查组检查督办湘水土地问题9年事件	新华网武汉7月3日电(记者...	99993	记者在追踪调查过程中了解到，美尔雅400亩地块旁边，还...	土地, 调查, 闲置, 等, 事件
北京燕郊上班族生活:每天朝六晚八5小时在路上	中新网12月11日电(房产频...	999897	据了解，往来于北京和燕郊的上班族，除了开车的高成本，...	北京, 每天, 元, 等, 燕
朝鲜民众称要掐死张成泽"扔进火炉"	国际在线专稿：据朝鲜《劳...	999900	他还表示："张成泽一党妄图敢挑战劳动党的权威，真想把他们扔进...	朝鲜, 表示, 金正恩, 民众, 张成
雅虎披露上季度31%营收来自与微软搜索合作	新浪科技讯北京时间12月1...	999909	SEC此前对雅虎表示："你们披露，从2011年至2012年，来自与...	雅虎, 合作, 微软, 搜索, 协议
林志颖惊入说调风波发新歌为自己打气	新浪娱乐讯台湾男星林志颖...	999915	今天早上，林志颖更在微博为自己打气。	林志颖, 微博, 自己, 节目, 因
英夫妇搬家种下圣诞小树35年后成参天大树(图)	中新网12月11日电(编外旗...	999921	在过去四年里，夫妇俩用这颗巨大的圣诞树已经募集了超过...	夫妇, 它, 颗, 用, 种
传朝鲜军队下达特别警戒令下步行动曾受猜测	【环球时报综合报道】"神...	999936	在朝鲜之外，忧虑的气氛越来越浓，美韩担心朝鲜对外更加...	朝鲜, 金正恩, 两, 将, 只
欧盟研发支出50强榜单：大众居首华为居第31位	新浪科技讯北京时间12月1...	999942	韩国的三星电子排名第二，研发支出约为83亿欧元。	美国, 日本, 制药, 德国, 汽车

图 8-34　汇总后的结果表

将第一条内容详细列出，我们可以对比查看抽取的关键句、关键词的效果，如表 8-4 所示。

表 8-4 新闻内容及抽取的关键句、关键词

title	湖北 4 个调查组检查督办浠水土地闲置 9 年事件
content	新华网武汉 7 月 3 日电（记者黄艳）新华社“新华视点”栏目 1 日播发的《土地“闲”九年，一场纠纷曝出乱象——湖北浠水土地闲置事件调查》报道，引起社会广泛关注。2 日，湖北省派出公安、纪检、国土、监察等四个小组再赴湖北浠水进行调查，共同督办这起事件。据介绍，调查小组主要针对冲突事件、党政干部有无牵涉违纪、土地闲置处理等进行调查和督办，另外组成一个总协调小组，对相关问题进行协调。记者在追踪调查过程中了解到，美尔雅 400 亩地块旁边，还有 3 块土地面临闲置处置问题。据浠水县滨江开发区管委会介绍，包括华新水泥、大唐荣耀工贸公司、新佳晶光电等公司在散花工业园投资建设的三块土地，也面临处置难的问题。浠水县国土局副局长饶伟表示，这三块土地“晒太阳”也超过了两年，根据国家相关土地政策，属于闲置土地。目前政府与企业协商处置面临困难，经济快速发展让土地资源越来越紧张，土地闲置的原因很复杂，如何运用法律手段推动闲置土地清理是当务之急。（原标题：湖北派出四部门调查督办浠水土地闲置事件）
doc_id	99993
abstract	记者在追踪调查过程中了解到，美尔雅 400 亩地块旁边，还有 3 块土地面临闲置处置问题。
keywords	土地, 调查, 闲置, 等, 事件

抽取的关键句，虽然没有像标题那样概括总结出问题，却是该条新闻中最核心的事件。抽取的关键词“土地”、“调查”、“闲置”、“事件”都比较准确，单词“等”不符合要求。如果希望进一步改进结果，可以通过将单词“等”加入停用词表，使用“停用词过滤”组件处理掉。

8.6 主题模型

本章所用的数据集是新闻文档的集合，每篇新闻的内容虽然不同，但是我们看过若干篇后，会发现有些新闻是在讲同一方面的事情，譬如都是关于电影的，或者都是关于意大利足球甲级联赛的，我们称之为主题。本节要讲的内容，是如何让计算机直接利用文本内容计算出其中有多少主题，为了控制主题的粒度，我们使用一个参数，即指定主题的总个数。

计算主题模型（Topic Model），有几个经典的方法：潜在语义分析（Latent Semantic Analysis，缩写为 LSA），概率潜在语义分析（Probabilistic Latent Semantic Analysis，缩写为 PLSA）和本节中重点介绍的潜在狄里克莱分布（Latent Dirichlet Allocation，缩写为 LDA）。这些方法的共同点是，我们会忽略每个文档中单词出现的顺序，文档是由出现了哪些单词和每个单词出现的个数，即词袋（bag of words）决定的。

8.6.1 LDA 模型

LDA（Latent Dirichlet Allocation）模型将贝叶斯估计的思想引入到了话题模型，使得参数具备了概率分布。LDA 是关于多个参数的分布，为了使读者容易理解此模型，我们先介绍简单的情形，即两个参数的情形，与此相关的有两个分布：二项分布和 Beta 分布。

二项分布（Binomial Distribution）

二项分布即重复 n 次独立的伯努利试验。在每次试验中只有两种可能的结果，而且两种结果发生与否相互对立，并且相互独立，与其他各次试验结果无关，事件发生与否的概率在每一次独立试验中都保持不变，则这一系列试验总称为 n 重伯努利实验。譬如：投硬币的实验，出现正面的概率为x，则出现负面的概率为$1-x$，那么，出现α_1次正面，α_2次负面的概率函数为：

$$P(\alpha_1,\alpha_2;x)=\binom{\alpha_1+\alpha_2}{\alpha_1}x^{\alpha_1}(1-x)^{\alpha_2}$$

Beta 分布

给定参数α_1，$\alpha_2>0$，取值范围为$[0,\ 1]$的随机变量x的概率密度函数为：

$$\begin{aligned}P(x;\alpha_1,\alpha_2)&=\frac{1}{B(\alpha_1,\alpha_2)}x^{\alpha_1-1}(1-x)^{\alpha_2-1}\\&=\frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}x^{\alpha_1-1}(1-x)^{\alpha_2-1}\end{aligned}$$

其中，$B(\alpha_1,\alpha_2)$为 Beta 函数，可以用 Gamma 函数$\Gamma(\cdot)$表示出来。如图 8-35 所示，列出三个 Beta 分布的概率密度分布图，参数的数值越大，在局部的概率密度就会越大。

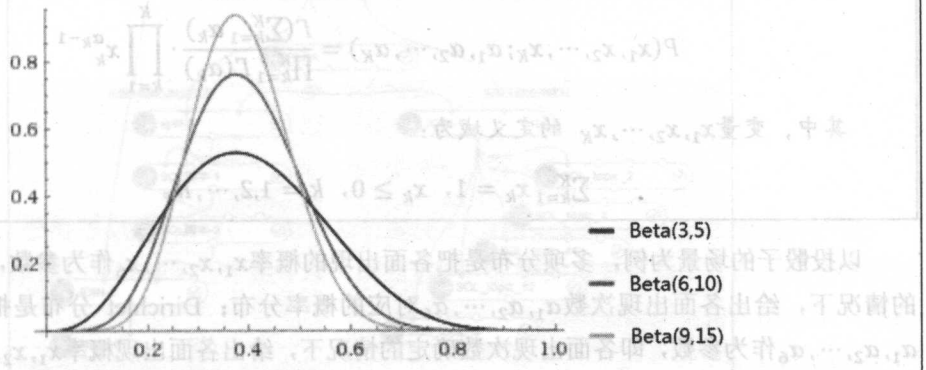


图 8-35 Beta 分布的概率密度分布图

还是以投硬币的场景为例，我们来看这两种分布。二项分布是把概率x作为参数，即概率x值确定的情况下，给出不同的正反面次数α_1和α_2对应的概率分布；Beta 分布是把α_1和α_2作为参数，即正反面出现次数确定的情况下，给出概率x对应的概率密度分布。

由贝叶斯理论，Beta 分布为二项分布的共轭先验分布。轭为驾车时搁在牛马颈上的曲木，可以起到束缚、控制的作用，使牛马行走同步。共轭可以理解为按照一定规律匹配的一对。

下面我们将“二项”推广为“多项”，将投硬币的场景扩展到投骰子的场景。在“二项”的时候，只给出一种结果的概率x，譬如：硬币为正面的概率，另外一种结果的概率可以用$1-x$表示。但在“多项”的情况，K种结果的概率要分别表示出来，$x_1, x_2, \cdots, x_K$，还要满足一个限制条件，所有结果的概率和为 1，即 $x_1 + x_2 + \cdots + x_K = 1$。

多项分布（Multinomial Distribution）

某随机实验如果有K个可能结局$A_1, A_2, \cdots, A_K$，譬如：骰子有六个面，各种结果出现的概率分别为$x_1, x_2, \cdots, x_K$，那么在$\alpha_1 + \alpha_2 + \cdots + \alpha_K$次试验结果中，$A_1$出现$\alpha_1$次、$A_2$出现$\alpha_2$次、……、$A_K$出现$\alpha_K$次的概率，即多项分布的概率密度函数为：

$$P(\alpha_1, \alpha_2, \cdots, \alpha_K ; x_1, x_2, \cdots, x_K) = \frac{(\sum_{k=1}^{K} \alpha_k)!}{\prod_{k=1}^{K} \alpha_k!} \cdot \prod_{k=1}^{K} x_k^{\alpha_k}$$

其中

$$\sum_{k=1}^{K} x_k = 1，\ x_k \geq 0，\ k = 1,2 \ldots K。$$

Dirichlet 分布

参数为K个正数$\alpha_1, \alpha_2, \cdots, \alpha_K$，其概率密度函数为：

$$P(x_1, x_2, \cdots, x_K; \alpha_1, \alpha_2, \cdots, \alpha_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \cdot \prod_{k=1}^{K} x_k^{\alpha_k - 1}$$

其中，变量$x_1, x_2, \cdots, x_K$ 的定义域为：

$$\sum_{k=1}^{K} x_k = 1，\ x_k \geq 0，\ k = 1,2, \cdots, K。$$

以投骰子的场景为例，多项分布是把各面出现的概率$x_1, x_2, \cdots, x_6$作为参数，即概率值确定的情况下，给出各面出现次数$\alpha_1, \alpha_2, \cdots, \alpha_6$对应的概率分布；Dirichlet 分布是把各面出现次数$\alpha_1, \alpha_2, \cdots, \alpha_6$作为参数，即各面出现次数确定的情况下，给出各面出现概率$x_1, x_2, \cdots, x_6$对应的概率密度分布。

LDA 主题模型如图 8-36 所示，详细内容请参考 David M. Blei、Andrew Y. Ng 和 Michael I.

Jordan 在 2003 年的论文（http://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf）。其中带阴影的圆圈表示可以观测到的变量，即文档内容，不带阴影的圆圈都是潜在变量。每篇文章的主题分布是按参数为α的 Dirichlet 分布采样得到的多项分布，每个主题的词分布是按参数为β的 Dirichlet 分布采样得到的多项分布。

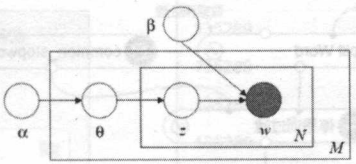
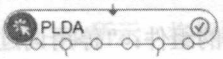


图 8-36　LDA 主题模型

8.6.2　新闻的主题模型

下面以新闻数据集 sina_news 为例，根据其文字内容计算主题模型。整体的流程如图 8-37 所示，核心组件为 PLDA 模型组件 ，围绕该组件有三个功能组件群：数据预处理、主题与原始分类的关系、查看主题的关键词。

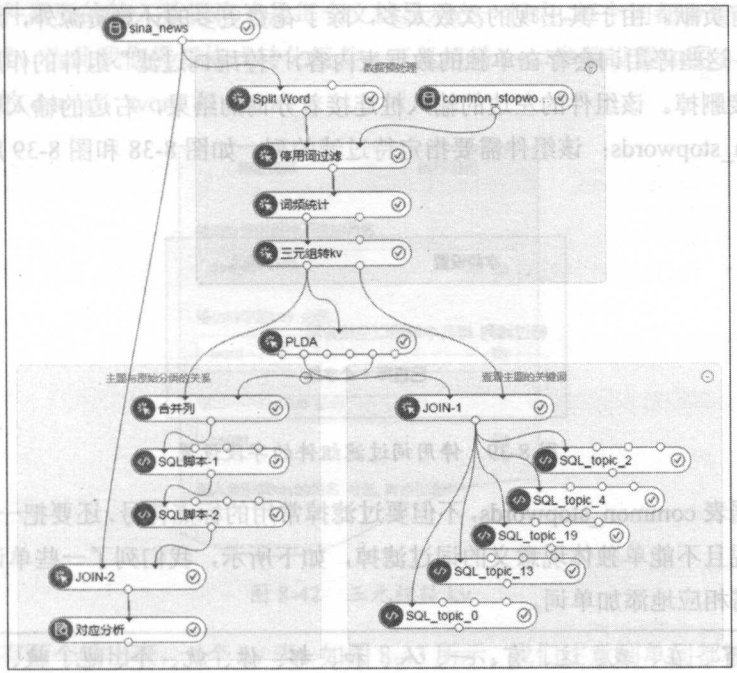


图 8-37　计算主题模型的整体流程

8.6.3 数据预处理

该项功能是由五个基础组件构成的，如图8-38所示，输入的原始数据首先会经过分词组件，该组件的设置与前几节相同，这里就不再多说。

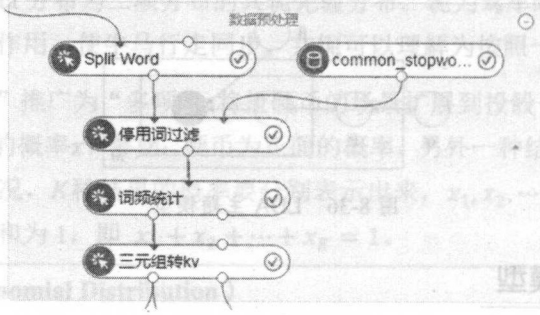
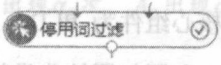


图 8-38 数据预处理

我们看到分词组件后接入了一个新的组件：“停用词过滤”组件 

。停用词是指标点符号和“的”，“了”等单词，在文档中大量出现，但是这些词在分析文档所表述的内容方面，没有贡献。由于其出现的次数太多，除了花费更多的计算资源外，还可能会影响内容分析的结果。这些停用词会存在单独的数据表内容，“停用词过滤”组件的作用是将文档中出现的停用词直接删掉。该组件的左边的输入桩连接着分词的结果，右边的输入桩连接着停用词数据表 common_stopwords；该组件需要指定待过滤的列，如图 8-38 和图 8-39 所示，选择新闻标题字段 title。

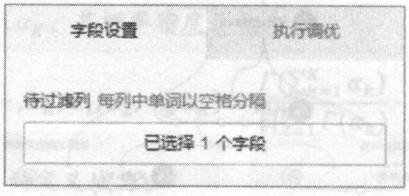


图 8-39 停用词过滤组件的字段设置

停用词数据表 common_stopwords，不但要过滤掉常用的标点符号，还要把一些助词、副词、代词等经常出现且不能单独体现意义的词过滤掉，如下所示，我们列了一些单词，根据不同的场景，用户还需相应地添加单词。

的、是、了、在、我、这、有、一、人、和、都、你、就、个、也、被、到、要、上、还、为、能、来、给、对、他、她、某某、我们、某、但、与、后、于、而、从、其、

“词频统计”组件在前面也介绍过，该组件在这里需要选择文档 ID 列为 doc_id，选择文档内容列为 title，及经过停用词过滤处理过的文档标题列 title，如图 8-40 所示。

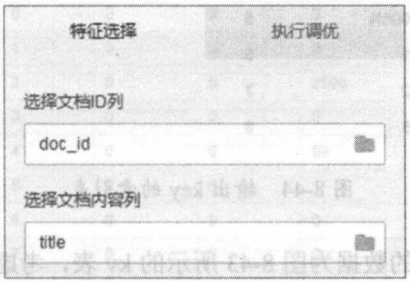


图 8-40 词频统计组件的参数设置

doc_id	word	count
135289	12	1
135289	名单	1
135289	大奖赛	1
135289	女排	1
135289	将	1
135289	担	1
135289	朱婷	1
135289	澳门	1

图 8-41 词频统计组件左边输出桩的输出结果

该组件左边输出桩的输出结果如图 8-41 所示，将标题中出现的不同单词逐行展开，并记录其所属文档的 ID 和出现的次数值。

但 PLDA 组件需要的输入数据格式是每个文档显示在一行内，每个词都带着其出现的次数，即 Key-Value 格式。为此我们需要格式转化组件：三元组转 kv。参数设置如图 8-42 所示，分别指定“三元组”的 doc_id，word 和 count。

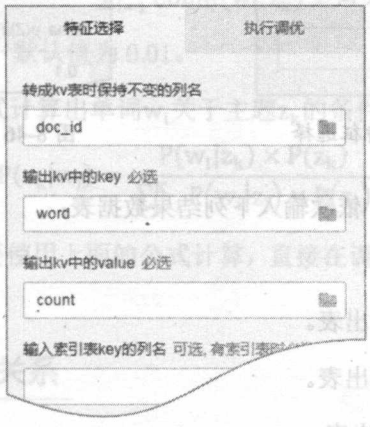


图 8-42 三元组转 kv

转换结果共有两个输出表，一个 kv 表，如图 8-43 所示，我们注意到单词都被数字替代了；而单词和数字的对应，即为另一个输出表：key 的索引表，如图 8-44 所示。

doc_id	key_value
207590	30973:1,35751:1,58691:1,72431:1,104111:1,1...
428868	32041:1,42282:1,58728:1,83082:1,86027:1,90...
647041	36139:1,41505:1,55594:1,70527:1,81533:1,82...
867623	4099:1,29733:1,31058:1,42393:1,44014:1,471...
135637	435:1,2504:1,5314:1,36024:1,38617:2,47741:1...
355384	39775:1,64107:1,65087:1,68949:1,77826:1,10...
574237	31058:1,31738:1,37153:1,46216:1,47260:1,59...
794369	31538:1,34647:1,40019:1,45351:1,47814:1,51...

图 8-43　输出 kv 表

key	key_id
(blob)	1
0	2
0%	3
0-1	4
0.00006%	5
0.001	6
0.002	7
0.005	8

图 8-44　输出 key 的索引表

下面我们可以进入 PLDA 组件的设置了，要处理的数据为图 8-43 所示的 kv 表，考虑到原始新闻类别的个数有十个左右，所以我们可以先试验划分 20 个主题，如图 8-45 所示，在“特征选择”选项卡中设置选择特征列为 key_value；如图 8-46 所示，在“参数设置”选项卡中设置主题个数为 20。

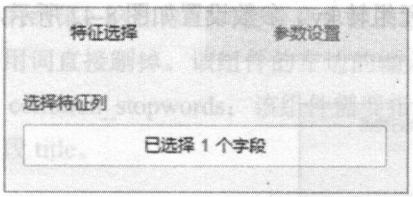


图 8-45　PLDA 组件的特征选择

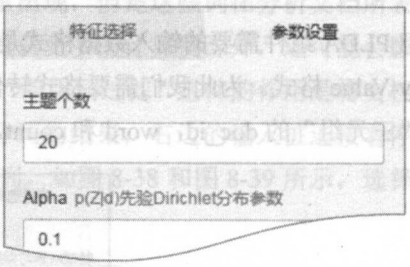


图 8-46　PLDA 组件的参数设置

PLDA 组件有 6 个输出桩，依次输入下列结果数据表：

1）主题-单词频数贡献表。

2）P(w/z)，P(单词|主题)输出表。

3）P(z/w)，P(主题|单词)输出表。

4）P(d/z)，P(文档|主题)输出表。

5）P(z/d)，P(主题|文档)输出表。

6）主题输出表。

其中的主题-单词频数贡献表是 LDA 计算结果中最基本的。通过查看 PLDA 组件的第一个输出桩获取，其内容如图 8-47 所示，每个单词对应一行，每个主题作为一列，表格中的数字即为该行所对应的单词对所在列的主题贡献的频数。

wordid	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	2508	0	358	0	0	0	51
3	0	0	0	0	0	0	0	0	0
4	0	0	99	0	25	0	0	0	0
5	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0

图 8-47 主题-单词频数贡献表

这个数据表是 LDA 算法标准的输出。下面，我们还会介绍由此表如何计算出 P(w|z)和 P(z|w)。

设文档的总数为N，单词的总数为M，主题的总数为K，频数贡献表中单词w_i对主题z_k的频数为$Count(w_i, z_k)$，则可以计算主题z_k关于单词w_i的条件概率：

$$P(w_i|z_k) = \frac{Count(w_i, z_k) + \beta}{\sum_{i=1}^{M} Count(w_i, z_k) + M \times \beta}$$

其中β为 LDA 的可选参数，默认值为 0.01。

进一步，可以使用下面公式计算出单词w_i关于主题z_k的条件概率：

$$P(z_k|w_i) = \frac{P(w_i|z_k) \times P(z_k)}{\sum_{j=1}^{K} P(w_i|z_j) \times P(z_j)}$$

在实际使用中，我们不需要使用上面的公式计算，直接在调用 LDA 函数时，指定需要的概率结果表即可。

8.6.4 主题与原始分类的关系

接下来，我们关注文档关于主题的概率 P(z|d)输出表，如图 8-48 所示。

docid	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8
0	0.008333...	0.008333...	0.008333...	0.008333...	0.008333...	0.008333...	0.008333...	0.175000...	0.008333.
1	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.344444...	0.0111111...	0.0111111
2	0.258333...	0.258333...	0.008333...	0.008333...	0.008333...	0.008333...	0.008333...	0.008333...	0.008333.
3	0.073333...	0.006666...	0.006666...	0.073333...	0.006666...	0.006666...	0.006666...	0.473333...	0.006666.
4	0.007142...	0.007142...	0.078571...	0.221428...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142.
5	0.01	0.01	0.110000...	0.01	0.01	0.01	0.01	0.01	0.01
6	0.14	0.006666...	0.006666...	0.006666...	0.006666...	0.206666...	0.006666...	0.473333...	0.006666.
7	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142.
8	0.009090...	0.009090...	0.009090...	0.009090...	0.009090...	0.009090...	0.009090...	0.009090...	0.009090.
9	0.091666...	0.008333...	0.008333...	0.008333...	0.175000...	0.008333...	0.091666...	0.091666...	0.008333.
10	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111...	0.0111111
11	0.65	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142...	0.007142.

图 8-48　P(z|d)输出表

第一列为 docid，即各文档的序号，与训练数据中文档的顺序是一致的；右边的各列 topic_0，…，topic_19 为其对应各个主题的概率。该输出表给出了每篇文档属于各主题的概率，从内容主题角度，赋予每篇文档一个向量值。我们可以根据不同的应用场景，解读、利用此概率向量数据做深入分析。

本节为了使读者更好地理解 LDA 主题模型结果的效果，我们直接将每个文档概率值最大的主题作为该文档的主题，然后和原始数据中每个文档在网站的类别进行对比，具体操作流程如图 8-49 所示。

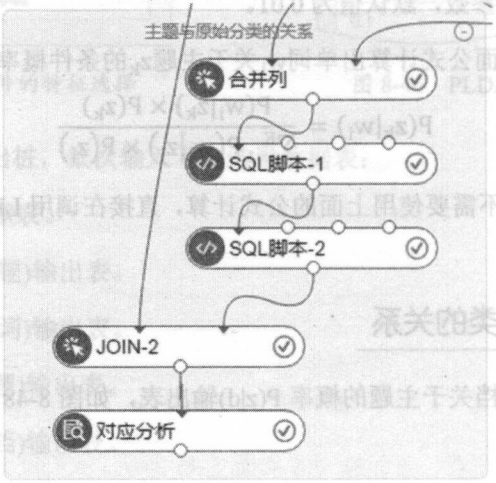


图 8-49　主题与原始分类关系的流程图

图 8-49 右上方的“合并列”组件的作用是，利用文档数据在 PLDA 组件处理前后，每篇文档的先后顺序没有改变，将原数据的文档 ID 字段 doc_id 加入进来，并保留对于各主题的概率值字段 topic_0，topic_2，…，topic_19，该组件的连接详见图 8-50，组件参数设置如图 8-51 所示。

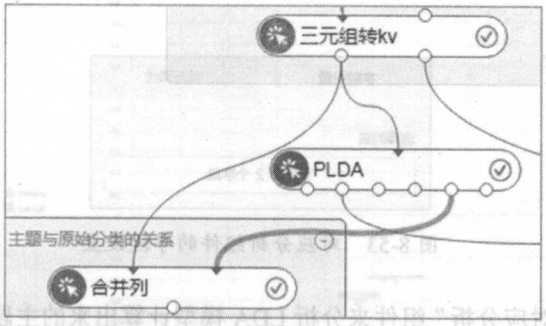


图 8-50　合并列组件的连接

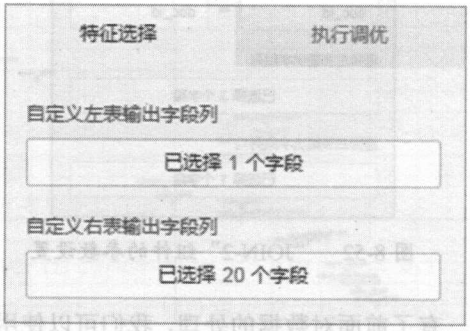


图 8-51　合并列组件的参数设置

随后的两个“SQL 脚本”组件，目的是计算出每个文档应该属于哪个主题。“SQL 脚本-1”组件的脚本如下，计算出每行中最大的概率值并保存在新字段 max_prob 中。

```
   select *, greatest(topic_0, topic_1, topic_2, topic_3, topic_4, topic_5,
topic_6, topic_7, topic_8, topic_9, topic_10, topic_11, topic_12, topic_13,
topic_14, topic_15, topic_16, topic_17, topic_18, topic_19) as max_prob from
${t1};
```

“SQL 脚本-2”组件的脚本如下，为节约篇幅略过了十几行有规律的代码。新增文档所属主题字段 topic_id，每个文档所属主题对应的概率值是该行中最大的概率值。

```
    select *,
    case when max_prob<=topic_1 then 1
    when max_prob<=topic_2 then 2
    when max_prob<=topic_3 then 3
    ……
    when max_prob<=topic_18 then 18
    when max_prob<=topic_19 then 19
    else 0 end AS topic_id
    from ${t1}
```

然后再用“JOIN-2”组件将新浪新闻分类字段与我们计算出的文档主题连接起来。如图 8-52 所示，使用内连接方式，将文档 ID 字段 doc_id 相同作为关联条件，在左表（原始数据）中选择字段 doc_id、title、category，在右表中选择 topic_id 字段。

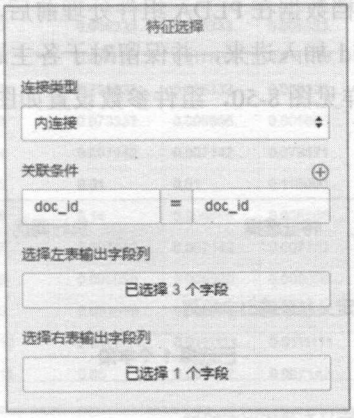


图 8-52 “JOIN-2”组件的参数设置

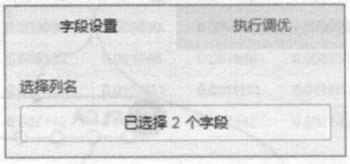


图 8-53 对应分析组件的字段设置

有了前面对数据的处理，我们可以使用“对应分析”组件来分析 LDA 模型计算出来的主题与新浪新闻分类之间的关系。组件设置如图 8-53 所示，选择主题字段 topic_id 和新浪新闻分类字段 category。“对应分析”组件的计算结果为一个数据表，如图 8-54 所示，将主题字段 topic_id 和新浪新闻分类字段 category 的每项内容都计算得出一个二维坐标，距离越近就越相似。

colname	value	rowpos	colpos
category	ent	-0.7459979954173978	1.4397175000492013
category	euro2012	-1.9756822434488819	-1.1371065024770186
category	finance	0.7113977168640367	-0.5939189602908357
category	green	0.4250454978516806	0.012175572706805755
category	health	-0.036927492649347436	0.30343570887659715
category	mil.news	0.3819816840422432	0.8951276623839114
category	news	0.3032983537335184	0.550615117141134
category	sports	-1.5064299957677099	-0.6456091580610885
category	tech	0.495094861431915	-0.24437645761925822
topic_id	0	-0.5697532186417223	1.3108544608392985
topic_id	1	-0.8043077043521019	0.816479713289796
topic_id	10	0.27792330541809346	0.10941229565682119
topic_id	11	0.761337268142221	-0.6492526497798793
topic_id	12	0.2649824589179473	0.0012075237963287023
topic_id	13	-0.8084546994193484	1.4160452970467983
topic_id	14	0.7055143905586807	-0.47997250763975524
topic_id	15	0.2621721138359825	0.6731821468486657
topic_id	16	0.8351033244591407	-0.8040540271554264

图 8-54 对应分析组件计算结果

再进一步可以使用对应分析组件的显示功能，在组件的右键菜单上选择“查看分析报告”，结果如图 8-55 所示。新浪新闻各分类用蓝色圆点标示；计算出的各主题位置用绿色方块标示。这里显示的是全部区域，可以通过下面的横条调整详细显示的区域。

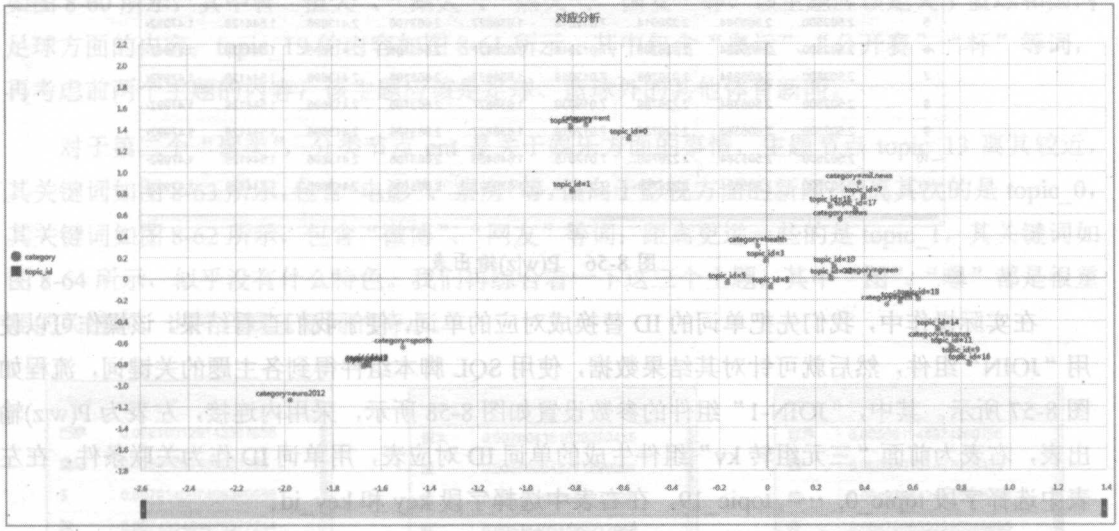


图 8-55 对应分析组件的分析显示

从图 8-55 上，我们可以看到所有分类节点和主题节点，明显地聚成了 3 个部分：

1）左下角区域，在分类节点 sports 和 euro2012 间有几个紧靠在一起的主题节点，调整下面的横条，可以看到共有 3 个主题节点，分别是 topic_2、topic_4 和 topic_19。

2）位于中部偏上区域，有一个分类节点 ent，即娱乐方面的新闻，主题节点 topic_13 离其较近，其次是 topic_0，稍远一些的是 topic_1。

3）其他的分类节点和主题节点都位于右侧中部，节点间还可以划分为更细的聚类。

为了进一步考察主题节点与其相邻的分类节点的关联情况，需要了解各主题节点的内容，这就用到了 LDA 的第二个输出表：主题关于单词的概率 P(w|z)输出表。如图 8-56 所示，第一列为 wordid，即单词的 ID；右边的各列 topic_0, … , topic_19 为该单词在各个主题上的概率，在某主题列的值越大，说明对该主题的贡献越大。我们再换个角度，将某主题列的值按从大到小排序，那么排在前面的单词应该是该主题最重要的单词，可以反映出该主题的内容。

wordid	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8
0	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
1	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
2	2.502500...	2.505384...	0.005584...	7.012018...	5.867019...	2.663106...	2.413696...	1.544128...	7.54776
3	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
4	2.502500...	2.505384...	2.204542...	7.012018...	4.097233...	2.663106...	2.413696...	1.544128...	1.47992
5	2.502500...	2.505384...	2.229014...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
6	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
7	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
8	2.502500...	2.505384...	2.226788...	7.019030...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
9	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992
10	2.502500...	2.505384...	2.226788...	7.012018...	1.640466...	2.663106...	2.413696...	1.544128...	1.47992
11	2.502500...	2.505384...	2.226788...	7.012018...	1.638827...	2.663106...	2.413696...	1.544128...	1.47992

图 8-56　P(w|z)输出表

在实际操作中，我们先把单词的 ID 替换成对应的单词，便于我们查看结果。该操作可以使用“JOIN”组件，然后就可针对其结果数据，使用 SQL 脚本组件得到各主题的关键词，流程如图 8-57 所示。其中，“JOIN-1”组件的参数设置如图 8-58 所示，采用内连接，左表为 P(w|z)输出表，右表为前面“三元组转 kv”组件生成的单词 ID 对应表，用单词 ID 作为关联条件，在左表中选择字段 topic_0, … , topic_19，在右表中选择字段 key 和 key_id。

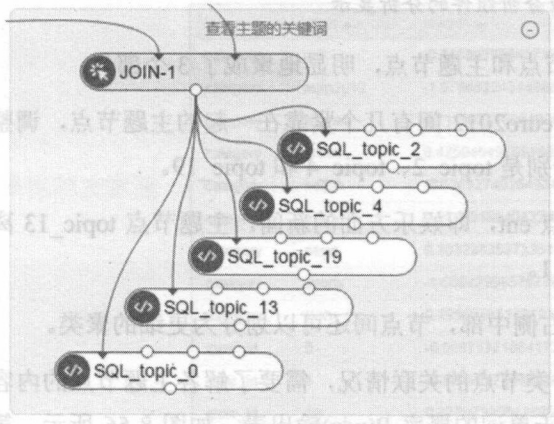


图 8-57　查看主题关键词的流程

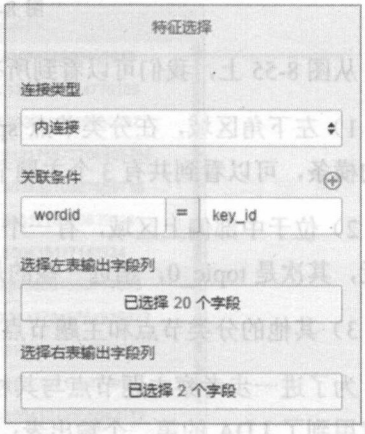


图 8-58　“JOIN-1”组件的参数设置

查看主题关键词的 SQL 脚本语句都是一个模式，以 topic_2 主题为例，其脚本为：

```
SELECT key, topic_2 FROM ${t1} ORDER BY topic_2 DESC LIMIT 100
```

有了查看主题关键词的能力，我们再回头看之前对应分析的结果。对于第一个“聚类”，分类节点 sports 和 euro2012 都是关于体育方面的事情，而 topic_2 的关键词如图 8-59 所示，其中有“巴萨”、“皇马”、“曼联”等，可以看出应该是关于国际足球方面的内容；topic_4 的关键词如图 8-60 所示，其中有“恒大”、“湖人”、“热火”、“国安”等，该主题应该是关于篮球和国内足球方面的内容；topic_19 的内容如图 8-61 所示，其中包含“奥运”、“公开赛”、“杯”等词，再考虑前两个主题的内容，该主题应该是足球、篮球外的其他体育新闻。

对于第二个“聚类”，分类节点 ent 是关于娱乐方面的事情，主题节点 topic_13 离其较近，其关键词如图 8-63 所示，包含“电影”、“票房”等，偏向于影视方面的新闻；距离其次的是 topic_0，其关键词如图 8-62 所示，包含“微博”、“网友”等词；距离更远一些的是 topic_1，其关键词如图 8-64 所示，似乎没有什么特色。我们再综合看一下这三个主题，其中“图”、“曝”都是很重要的关键词，也符合娱乐新闻的特点。

key	topic_2
巴萨	0.008190129142357656
皇马	0.00804761469901985
3	0.007914007408390659
球	0.007486464078377244
曝	0.007192528038993022
年	0.0068562830242428876
宣布	0.006397564659749328
曼联	0.006344121743497651
新	0.006076907162239267
甲	0.005987835635153139
签	0.005932165930724309
0	0.005584786975088409
AC米...	0.005551385152431111
帅	0.005317572393830025
国米	0.005172831162315067

图 8-59　topic_2

key	topic_4
恒大	0.007004351038260415
没	0.005590042771599097
打	0.005422882350742255
队	0.005354051589212968
?	0.005288498482994598
湖人	0.005026286058121121
足	0.0048083219799450435
战	0.004462529344643145
热火	0.004405170376702072
大	0.00438878210014748
两	0.0042822583025426296
场	0.004154429745416809
国安	0.004105264915753032
将	0.003990546979870886
球员	0.00391843856303068

图 8-60　topic_4

key	topic_19
世界	0.005091948874860116
四	0.005030709427919459
位	0.0049785424916366765
杯	0.0047086352995648895
将	0.004640591469630825
获	0.004629250831308482
站	0.004468213767131197
前	0.0044523368734799154
奥运	0.004363879894565632
三	0.004359343639236695
公开赛	0.00435027112857882
决赛	0.004245937256013255
3	0.004234596617690911
领先	0.003987370702263812
年	0.003901181851013997

图 8-61　topic_19

key	topic_0
图	0.035150127143301124
曝	0.013328321158125286
照	0.006644141824822
不	0.0063563542458890615
称	0.006201199203333912
岁	0.005870869112732627
否认	0.005598096537917928
女	0.005355353971339711
晒	0.005070068893093146
曝光	0.005055053888974905
传	0.004734733801119113
网友	0.004697196290823513
遭	0.004351851196103987
曾	0.004249248667962679
微博	0.004159158643253237

图 8-62　topic_0

key	topic_13
图	0.016051409723853344
将	0.012414742217327305
新	0.006314100711814419
大	0.0060136803525796595
曝	0.005555144014800289
电影	0.005534061884327674
获	0.004859433709204003
之	0.0047909167851680045
搜	0.004258592990734483
演	0.004082030148026335
首	0.0038395856475912653
组图	0.0037209986636828076
曝光	0.0036709286038103476
再	0.00361031747870158
票房	0.0035866000819198885

图 8-63　topic_13

key	topic_1
不	0.02423458929691112
图	0.01022698309910032
好	0.007691533721138501
称	0.007047649837604719
没	0.0070225959899964
谈	0.00598536669901202
想	0.005845065152405437
很	0.005842559767644605
看	0.004968180486114294
曝	0.0046499966214886505
做	0.004509695074882068
太	0.0043568666044713255
自	0.0041589412083656106
让	0.0041589412083656106
爱	0.004096306589344814

图 8-64　topic_1

8.7　单词映射为向量

Word2Vec，即 Word to Vector，就是将单词映射为向量，将文本内容的处理变成向量间的计算。譬如，单词语义上的相似性比较，相当于单词对应向量间“距离”的比较，这里“距离”用引号，表示我们可以使用不同的距离定义，欧式距离、余弦距离等。

整体的流程如图 8-65 所示，最开始的三个组件实现了对原始文档的分词处理；去掉标点符号等停用词；通过“词频统计”组件，将文档数据变为“Word2Vec”组件需要的输入处理格式。针对向量化后的结果，可以通过计算语义向量距离，了解与某个词“距离”最近的有哪些词；也可以按照向量值，对单词进行聚类，看看每个小类别里有哪些单词。

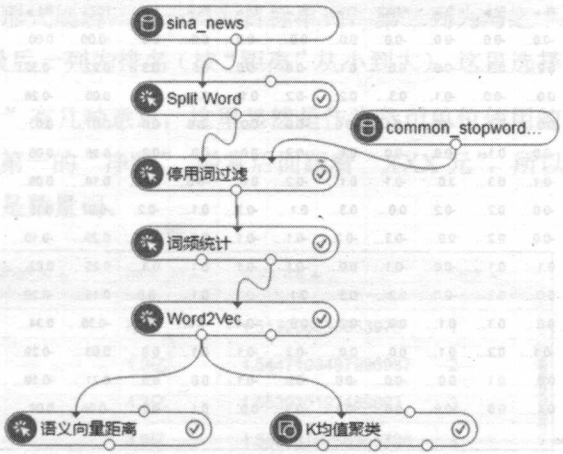


图 8-65　向量化的整体流程

其中，“Word2Vec”组件的字段设置如图 8-66 所示，在“字段设置”选项卡中选择单词列 word；并在“参数设置”选项卡中设置单词的特征维度为 100，其他参数使用默认值即可，如图 8-67 所示。

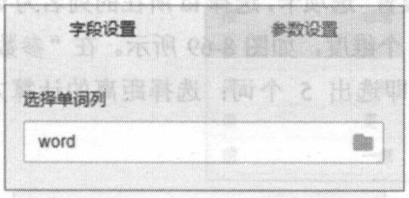


图 8-66　Word2Vec 组件的字段设置

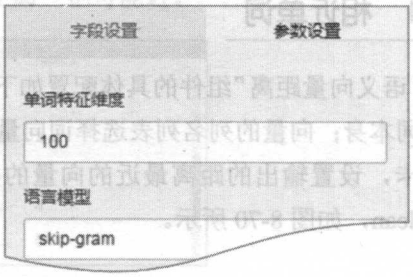


图 8-67　Word2Vec 组件的参数设置

“Word2Vec”组件的计算结果如图 8-68 所示，第一列显示的单词，后面 f0 列到 f99 列为该单词对应的向量值。

word	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14
</s>	0.0...	-0.0...	-0.0...	-0.0...	-0.0...	0.0...	0.0...	-0.0...	6.6...	0.0...	-0.00...	0.00...	-0.00...	-0.00...	-6.65...
不	-0.1...	0.2...	0.3...	-0.0...	0.0...	0.1...	-0.0...	-0.0...	0.1...	0.3...	0.22...	-0.32...	0.25...	0.17...	-0.24...
公司	-0.0...	0.0...	-0.0...	-0.1...	0.3...	0.2...	-0.2...	0.1...	0.1...	0.1...	0.00...	-0.26...	-0.14...	0.02...	-0.05...
将	0.1...	0.0...	0.1...	-0.0...	-0.1...	-0.0...	-0.0...	0.0...	-0.0...	0.0...	0.07...	0.03...	0.10...	-0.11...	-0.22...
中	-0.1...	-0.0...	0.1...	-0.0...	-0.0...	0.1...	-0.2...	0.0...	0.0...	0.2...	0.16...	0.06...	0.05...	-0.04...	-0.06...
中国	-0.0...	-0.1...	0.3...	3.0...	-0.1...	0.1...	-0.2...	0.0...	-0.0...	0.0...	0.18...	0.06...	0.26...	-0.00...	0.10...
批	0.1...	-0.0...	0.2...	-0.2...	0.0...	0.3...	0.1...	-0.3...	0.1...	-0.2...	-0.27...	0.13...	-0.13...	-0.11...	-0.18...
多	-0.0...	-0.0...	0.2...	0.0...	-0.2...	-0.2...	-0.1...	-0.1...	0.3...	-0.0...	0.29...	-0.10...	0.04...	0.11...	-0.46...
说	0.0...	0.1...	0.1...	-0.0...	-0.1...	0.0...	-0.1...	-0.1...	0.1...	0.3...	0.25...	0.02...	0.13...	-0.06...	0.07...
元	-0.1...	0.0...	0.3...	-0.0...	0.2...	0.3...	0.1...	-0.1...	0.1...	-0.0...	0.15...	-0.20...	0.12...	-0.01...	-0.29...
碌	-0.2...	0.0...	0.3...	0.1...	0.0...	0.2...	0.5...	-0.4...	-0.0...	-0.2...	-0.38...	0.34...	-0.01...	-0.06...	-0.44...
等	0.0...	-0.1...	0.2...	0.1...	0.0...	0.0...	-0.2...	-0.1...	0.1...	0.0...	0.03...	-0.29...	0.10...	0.26...	-0.01...
很	0.1...	0.0...	0.1...	0.0...	-0.0...	-0.0...	-0.0...	-0.1...	0.0...	0.5...	0.21...	-0.19...	0.52...	0.38...	-0.16...
两	-0.2...	0.4...	0.0...	-0.0...	-0.0...	-0.1...	-0.3...	0.0...	0.1...	0.2...	-0.06...	0.09...	0.04...	-0.23...	-0.16...

图 8-68　Word2Vec 组件的计算结果

单词对应到向量值，单词间便有了“距离”的概念。从单词个体的角度，可以找到与之“距离”最近的单词，即最“相似”的单词；从群体的角度，可以聚成若干个小类别，每个类别包含的单词的词间“距离”较近，也比较“相似”。后面的两个小节将具体讲述相近单词和单词聚类。

8.7.1　相近单词

“语义向量距离”组件的具体配置如下：在“字段设置”选项卡，选择 id 所在的列名为 word，及单词本身；向量的列名列表选择词向量的全部 100 个维度，如图 8-69 所示。在“参数设置”选项卡，设置输出的距离最近的向量的数目为 5，即选出 5 个词；选择距离的计算方式为 euclidean，如图 8-70 所示。

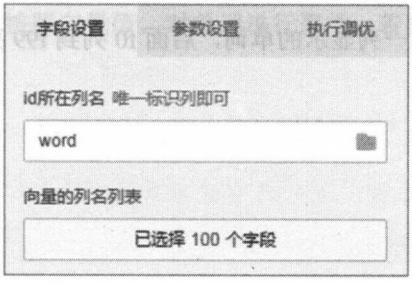


图 8-69　“语义向量距离”组件的字段设置

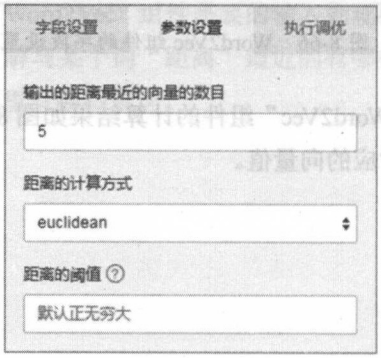


图 8-70　“语义向量距离”组件的参数设置

计算结果以数据表的形式展现，第一列为目标单词，第二列为与之“距离”较近的单词，第三列为“距离”的值，最后一列为排名（按“距离”从小到大）。这里选择了几个典型的单词。

如图 8-71 所示，“元”有几种意思，这里显然是作为货币单位使用的；这里“距离”近不代表意思相近，例如排名第一的“净赚”，经常后面跟着“XXX 元”，所以才“距离”近，而随后的四个“距离”近的都是数量词。

original_id ▲	near_id ▲	distance ▲	rank ▲
元	净赚	1.513206638653975	1
元	1.9亿	1.5447103487996987	2
元	4.3亿	1.552920193489893	3
元	3.9亿	1.5583879256355786	4
元	2.8亿	1.5598111936056542	5

图 8-71　与单词“元”相近的单词

如图 8-72 所示，可以看出前两个词都是“很”的同义词，随后的两个词“比较”、“更”，关联就弱了一些，而第 5 个词“一点”，与“很”就差更多了，甚至像是反义词。我们可以更清晰地理解“距离”，同义词“距离”近，反义词的“距离”也可能近。

original_id ▲	near_id ▲	distance ▲	rank ▲
很	非常	0.9316219...	1
很	太	1.0403879...	2
很	比较	1.1418719...	3
很	更	1.2062492...	4
很	一点	1.2072073...	5

图 8-72　与单词“很”相近的单词

下面，我换一个距离定义看看，如图 8-73 所示，选择距离的计算方式为 cosine，即余弦距离。

从结果集中找个例子看看，单词“两”，如图 8-74 所示，“三”、“四”、“六”、“五”依次为与之“距离”较近的单词；单词“几”表示不确定的数目，“两”在有些语境里，也不是确定的数目，它们也可以看作是相近的词。

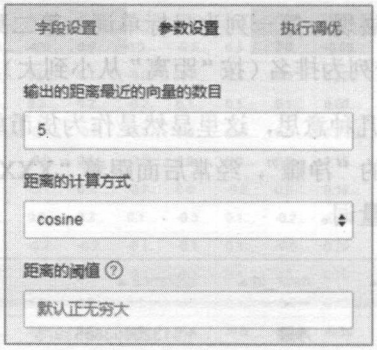


图 8-73 “语义向量距离”组件的参数设置

original_id	near_id	distance	rank
两	三	0.1256223...	1
两	四	0.1578440...	2
两	六	0.1800341...	3
两	几	0.2167491...	4
两	五	0.2548505...	5

图 8-74 与单词“两”相近的单词

我们更关心不同距离的计算方式对计算结果的影响。下面将进行对比分析。

对于单词“说”，欧式距离（Euclidean）的计算结果如图 8-75 所示；余弦距离（Cosine）的计算结果如图 8-76 所示。在不同距离定义方式下，“知道”一词始终排在第一，在余弦距离下，与其动词的意思相近的“讲”和“谈”都出现了，但在欧式距离下，这两个词排名都没有进前五。

original_id	near_id	distance	rank
说	知道	1.0102247063452177	1
说	原话	1.0433025546974142	2
说	实话实说	1.0446003441808407	3
说	说和	1.0772313472405493	4
说	意思	1.0823357406006942	5

图 8-75 欧式距离（Euclidean）

original_id	near_id	distance	rank
说	知道	0.1476631...	1
说	讲	0.1681493...	2
说	看来	0.1847001...	3
说	谈	0.1875879...	4
说	却说	0.1923427...	5

图 8-76　余弦距离（Cosine）

对于单词“市场”，欧式距离的计算结果如图 8-77 所示；余弦距离的计算结果如图 8-78 所示。在不同距离定义方式下，“金融化”和“整体而言”都出现了。

original_id	near_id	distance	rank
市场	金融化	1.256358522829559	1
市场	整体而言	1.3163838985849325	2
市场	联动性	1.3518281284107458	3
市场	泡沫化	1.3755966878815622	4
市场	车市	1.3798319022768866	5

图 8-77　欧式距离（Euclidean）

original_id	near_id	distance	rank
市场	金融化	0.2204331...	1
市场	活跃性	0.2331473...	2
市场	整体而言	0.2444738...	3
市场	利率化	0.2484193...	4
市场	趋向	0.2704267...	5

图 8-78　余弦距离（Cosine）

8.7.2　单词聚类

使用“K 均值聚类”组件实现，使用“Word2Vec”组件的结果之后，为了更好地理解聚类结果，我们还使用了两个“SQL 脚本”组件显示某个具体聚类的内容，整体流程如图 8-79 所示。

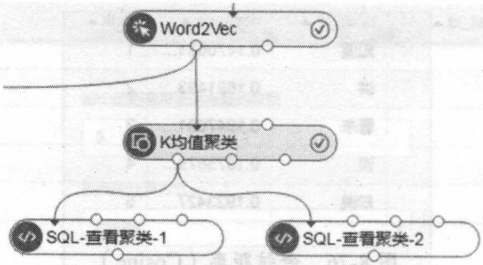


图 8-79　单词聚类实验流程

设置“K 均值聚类”组件，在“字段设置”选项卡中，选择特征列为单词向量的 100 个维度，附加列选择单词 word 字段，如图 8-80 所示；在“参数设置”选项卡中，设置聚类数为 1000，如图 8-81 所示。

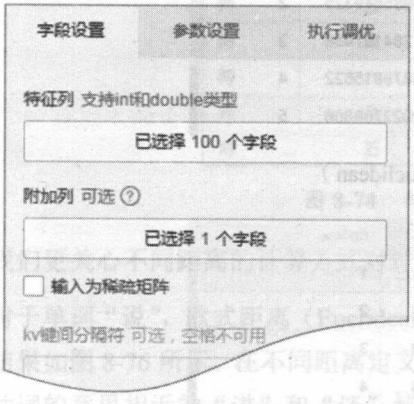


图 8-80　“K 均值聚类”组件的字段设置



图 8-81　“K 均值聚类”组件的参数设置

执行结果输出两张数据表和一个模型：

1）聚类结果表，和输入表对应，并指明聚类后每条 record 所属的类号。

2）聚类中心点模型。

3）聚类统计表，统计各个聚类包含的点的数目。

“K 均值聚类”组件的第一个输出桩输出的聚类结果表，内容如图 8-82 所示，每个单词都对应着一个聚类索引号。

word	cluster_index
不	946
公司	890
将	952
中	968
中国	396
拢	505
多	493
说	983
元	837

图 8-82　输出的聚类结果

这里我们可以随机挑选两个聚类，查看效果。

1）组件 

的 SQL 脚本如下，查看索引号 710 的聚类情况。

```
select * from ${t1} where cluster_index=710
```

该聚类中部分单词如图 8-83 和图 8-84 所示，显示是关于国际货币方面的。

word	cluster_index
美元	710
人民币	710
欧元	710
英镑	710
日元	710
汇率	710
约合	710
升值	710
贬值	710

图 8-83　聚类（#710）的结果（一）

word	cluster_index
韩元	710
逆差	710
盈余	710
中间价	710
即期	710
卢比	710
卢布	710
克朗	710
比索	710

图 8-84 聚类（#710）的结果（二）

2）组件 SQL-查看聚类-2 的 SQL 脚本如下，查看索引号 890 的聚类情况。

```
select * from ${t1} where cluster_index=890
```

该聚类中部分单词如图 8-85 和图 8-86 所示，显示是关于“公司”方面的。

word	cluster_index
公司	890
有限	890
股份	890
集团	890
子	890
控股	890
简称	890
100%	890
全资	890

图 8-85 聚类（#890）的结果（一）

word	cluster_index
华星	890
国电	890
浙能	890
纸业	890
化纤	890
中联	890
微电子	890
塑胶	890
工贸	890

图 8-86 聚类（#890）的结果（二）

8.8 组件使用小结

本节将前面介绍的各种文本组件的调用关系及所需的数据依赖，用一张关系图描绘了出来，如图 8-87 所示。

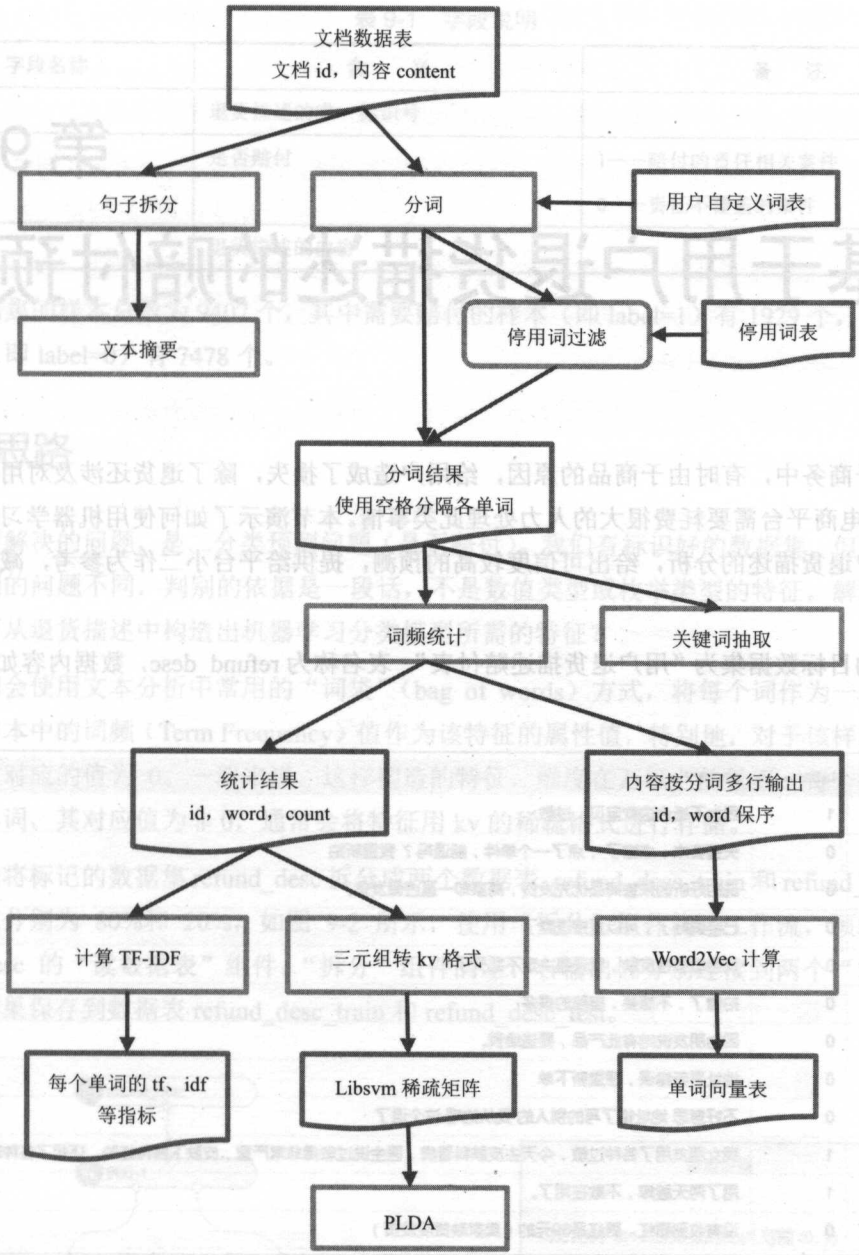


图 8-87　常用文本组件调用关系

第9章 基于用户退货描述的赔付预测

在电子商务中，有时由于商品的原因，给用户造成了损失，除了退货还涉及对用户进行赔付的问题。电商平台需要耗费很大的人力处理此类事情。本节演示了如何使用机器学习的办法，通过对用户退货描述的分析，给出可信度较高的预测，提供给平台小二作为参考，减少人力成本。

我们的目标数据集为“用户退货描述赔付表”，表名称为 refund_desc，数据内容如图 9-1 所示。

doc_id	label	content
0	1	因为不适合这款宝贝，过敏
1	0	买整套来，点错了，点了一个单件，能退吗？我重新拍
2	0	说的好听的预售商品优先发货，商家却一直点假发货。
3	0	已经说好了，可以直接退款了
4	0	我是敏感性皮肤，我还是决定不乱买了
5	0	拍错了，不想要，擦脸的很多
6	0	因为朋友说她有此产品，要送给我。
7	0	地址更新错误，想重新下单
8	0	不好意思 地址错了写的别人的 我从拍吧 这个退了
9	1	我女朋友用了各种过敏，今天去皮肤科看病，医生说过敏得非常严重，皮肤下各种红肿，还起了各种斑，...
10	1	用了两天脸痒，不敢在用了。
11	0	没有收到腮红，腮红是69元的（卖家缺货没发货）

图 9-1　数据表 refund_desc 的内容

该数据集有三个数据字段，其含义及说明如表 9-1 所示。

表 9-1　字段说明

字段名称	含　义	备　注
doc_id	退货描述的唯一标识号	
label	是否赔付	1——赔付的责任相关案件 0——责任不覆盖的案件
content	退货描述的内容	

数据集的样本总数为 9407 个，其中需要赔付的样本（即 label=1）有 1929 个，不需要赔付的样本（即 label=0）有 7478 个。

9.1　思路

所要解决的问题，是二分类预测问题（是否赔付）。我们有标识好的数据集，但是和前面章节所遇到的问题不同，判别的依据是一段话，不是数值类型或枚举类型的特征。解决问题的关键：如何从退货描述中构造出机器学习分类模型所需的特征？

我们会使用文本分析中常用的“词袋”（bag of words）方式，将每个词作为一个特征，而该词在样本中的词频（Term Frequency）值作为该特征的属性值，特别地，对于该样本中没有出现的词，对应的值为 0。一般来说，这样构造的特征，维度在万以上的量级，每个样本只包含部分的单词，其对应值为非 0，通常会将特征用 kv 的稀疏格式进行存储。

首先将标记的数据集 refund_desc 拆分成两个数据表：refund_desc_train 和 refund_desc_test，数据占比分别为 80%和 20%，如图 9-2 所示，使用“拆分”组件构建工作流，顶端是数据表 refund_desc 的“读数据表”组件；“拆分”组件的左、右输出桩分别连接到两个“写数据表”组件，结果保存到数据表 refund_desc_train 和 refund_desc_test。

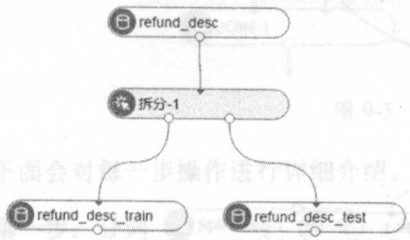


图 9-2　数据拆分

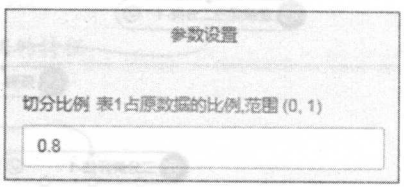


图 9-3　“拆分”组件的参数设置

其中，“拆分”组件需要设置左输出桩的数据占比，参数设置界面如图 9-3 所示，这里我们设置为 0.8，即 80%的数据；其余的数据会输出到右边的输出桩。

基于此训练数据集和测试数据集，整体的流程图如图 9-4 所示，我们需要重点关注以下三点：

1）使用常用的文本转换特征方式：分词、停用词过滤、词频统计、TF-IDF 计算、转为 kv 格式。

2）需要将标记的数据集，分为训练和测试两部分，使用训练数据集得到分类模型，通过测试数据集，检验模型的鲁棒性。在预处理的阶段，将原始数据 refund_desc 拆分为两个数据表 refund_desc_train 和 refund_desc_test。

3）在这个应用中，我们更关注需要赔付的样本，希望对于那些需要赔付的样本预测出的比例更大，即希望分类模型有更高的召回率。

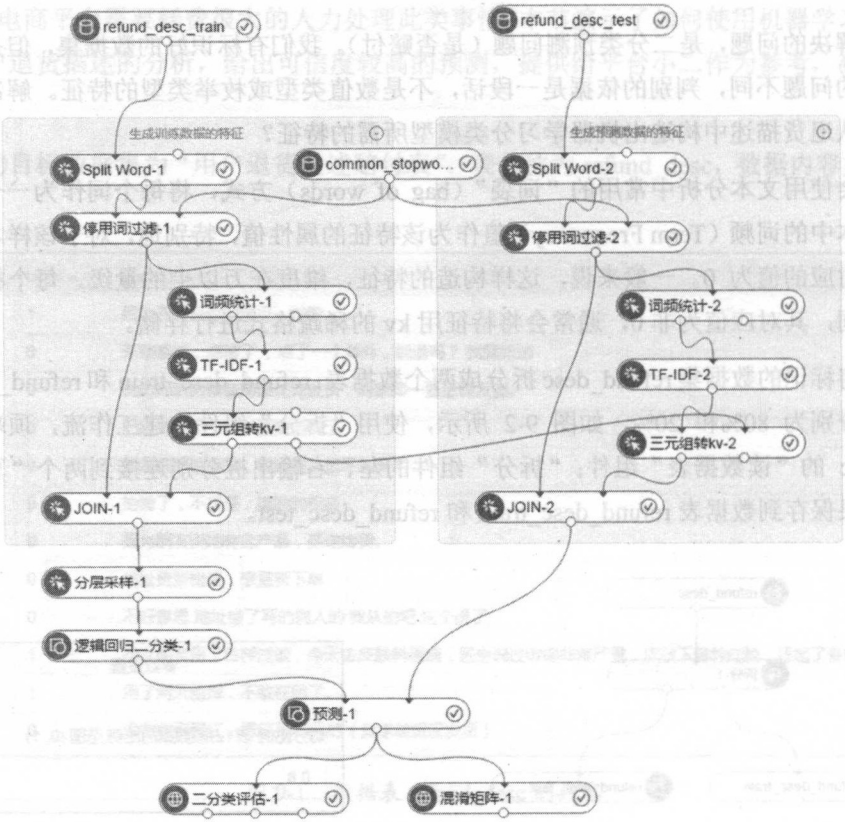


图 9-4　完整流程

我们需要关注两个重要的功能区“生成训练数据的特征”和“生成预测数据的特征”。

- 相同点：基本流程一致；都使用同一个停用词表“common_stopwords”。
- 差异点：在“三元组转 kv”组件的使用上，根据训练数据会产生“key 的索引表”，即组件的第二个输出桩；为了保证预测产生的特征索引的一致性，需要通过组件的第二个输入桩，引用该“key 的索引表”。

9.2　训练集的特征生成

整体流程如图 9-5 所示，共有六个处理组件，需要使用停用词表“common_stopwords”，有两处“向外部的输出”。

1）“JOIN”组件的输出是特征生成的结果。

2）“三元组转 kv”组件输出 key 的索引表。测试集需要使用此所以表，使其特征索引与训练集一致。

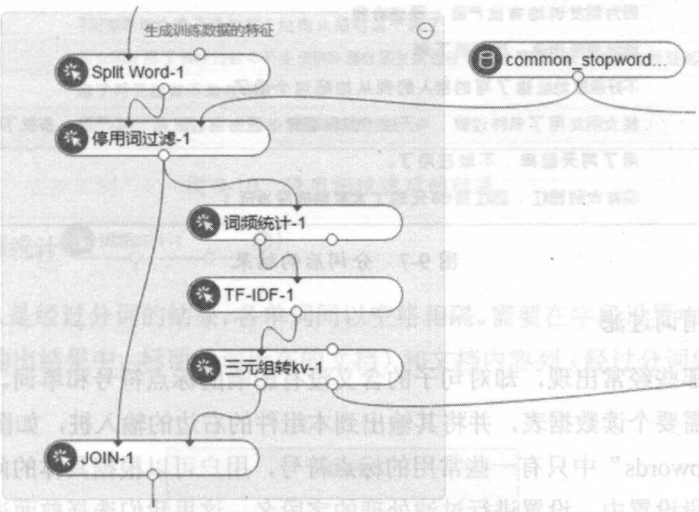
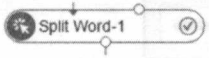


图 9-5　生成训练数据的特征

下面会对每一步操作进行详细介绍。

第一步：分词 

需要在“字段设置”选项卡中，设置需要分词处理的字段名，这里我们选择退货描述的字

段“content”，其余的参数使用默认设置，如图 9-6 所示。

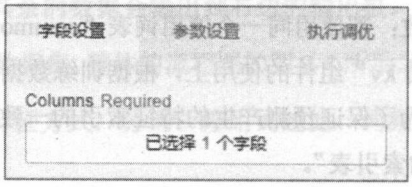


图 9-6　分词组件的字段设置

输出结果如图 9-7 所示，整句话被分成了词的粒度，单词之间用空格符号作为间隔。

doc_id ▲	label ▲	content ▲
0	1	因为 不 适合 这 款 宝贝 ， 过敏
1	0	买 整套 来 ， 点错 了 ， 点 了 一个 单件 ， 能 退 吗 ？ 我 重新 拍
2	0	说 的 好听 的 预售 商品 优先 发货 ， 商家 却 一直 虚假 发货 。
3	0	已经 说 好 了 ， 可以 直接 退款 了
4	0	我 是 敏感性 皮肤 ， 我 还是 决定 不 乱 买 了
5	0	拍 错 了 ， 不 想 要 ， 操 脸 的 很多
6	0	因为 朋友 说 她 有 此 产品 ， 要 送给 我 。
7	0	地址 更新 错误 ， 想 重新 下单
8	0	不好意思 地址 错 了 写 的 别人 的 我 从 拍 吧 这个 退 了
9	1	我 女朋友 用 了 各种 过敏 ， 今天 去 皮肤科 看病 ， 医生 说 过敏 得 非常 严重 ， 皮肤 下 各种 红肿 ， 还 ...
10	1	用 了 两 天 脸 痒 ， 不敢 在 用 了 。
11	0	没有 收到 腮红 ， 腮红 是 69 元 的 （ 卖家 缺货 没 发货 ）

图 9-7　分词后的结果

第二步：停用词过滤

主要是去掉那些经常出现，却对句子的含义没有影响的标点符号和单词。此操作需要指定停用词表，所以需要个读数据表，并将其输出到本组件的右边的输入桩，如图 9-8 所示。数据表“common_stopwords”中只有一些常用的标点符号，用户可以根据具体的问题，扩充停用词表。还需要在字段设置中，设置进行过滤处理的字段名，这里我们选择前面进行过分词处理的退货描述字段“content”，如图 9-9 所示。

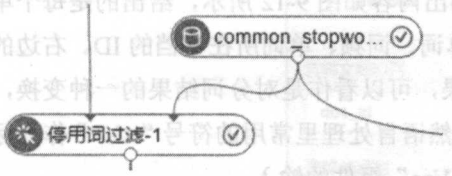


图 9-8　停用词过滤组件的连接

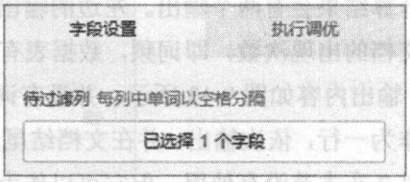


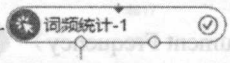
图 9-9　停用词过滤组件的字段设置

处理结果如图 9-10 所示，去掉了逗号，句号，括号等。

doc_id	label	content
0	1	因为不适合这款宝贝过敏
1	0	买整套来点错了点了一个单件能退吗？我重新拍
2	0	说的好听的预售商品优先发货商家却一直虚假发货
3	0	已经说好了可以直接退款了
4	0	我是敏感性皮肤我还是决定不乱买了
5	0	拍错了不想要擦脸的很多
6	0	因为朋友说她有此产品要送给我
7	0	地址更新错误想重新下单
8	0	不好意思地址错了写的别人的我从拍吧这个退了
9	1	我女朋友用了各种过敏今天去皮肤科看病医生说过敏得非常严重皮肤下各种红肿还起了各种...
10	1	用了两天脸痒不敢在用了
11	0	没有收到腮红腮红是69元的卖家缺货没发货

图 9-10　停用词过滤后的结果

第三步：词频统计



该组件的输入是经过分词的结果，各单词间以空格相隔。需要在字段设置中设置两个参数：文档 ID（用来在输出结果中，标明单词所在的文档）和文档内容列（经过分词处理过的内容）。如图 9-11 所示。

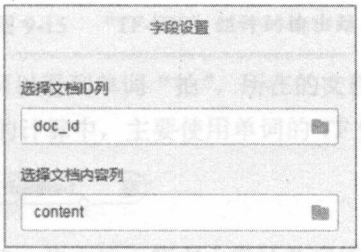


图 9-11　词频统计组件的参数设置

计算结果会有两个输出。左边的输出桩 1，输出内容如图 9-12 所示，给出的是每个单词在所在文档的出现次数，即词频，数据表有三列：单词、词频、单词所在文档的 ID。右边的输出桩 2，输出内容如图 9-13 所示，并没有词频的结果，可以看作是对分词结果的一种变换，将每个词作为一行，依次输出，并在文档结尾，使用自然语言处理里常用的符号“</s>”作为标识。输出桩 2 在本节没有使用，但它可以作为“Word2Vec”组件的输入。

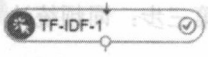
doc_id	word	count
0	不	1
0	因为	1
0	宝贝	1
0	款	1
0	过敏	1
0	这	1
0	适合	1
1	一个	1
1	买	1
1	了	2
1	单件	1
1	吗	1
1	我	1
1	拍	1

图 9-12　输出桩 1：三元组输出

doc_id	word
0	因为
0	不
0	适合
0	这
0	款
0	宝贝
0	过敏
0	</s>
1	买
1	整套
1	来
1	点
1	错
1	了

图 9-13　输出桩 2：多行输出

第四步：TF-IDF（Term Frequency - Inverse Document Frequency）



TF-IDF 组件在“8.3 单词的区分度”一节里有详细的介绍，这里就不再重复。该组件的输入是“词频统计”组件的输出桩 1 的结果，需要在字段设置中，配置三个参数，文档 ID、单词列和单词计数列，具体设置如图 9-14 所示。

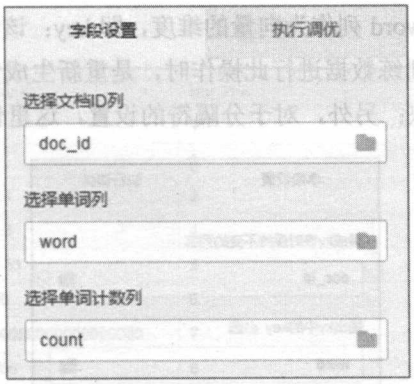


图 9-14　“TF-IDF”组件的字段设置

计算结果如图 9-15 所示。可以看到，第一列为文档标号，第二列为我们关注的单词，第三列total_word_cnt为该单词在该文档中出现的次数，第四列是包含该单词的文档数目doc_count，第五列为全部的文档数目 total_doc_count，最后三列为 TF、IDF 和 TF-IDF 指标。

id	word	count	total_word_count	doc_count	total_doc_count	tf	idf	tfidf
100	和	1	10	250	7521	0.1	3.4039934689641567	0.3403993468964157
100	商家	1	10	161	7521	0.1	3.84405002184194	0.38440500218419404
100	拍	2	10	1141	7521	0.2	1.8857940369643273	0.3771588073928655
100	联系	1	10	328	7521	0.1	3.1324407784422594	0.313244077844226
100	过	1	10	328	7521	0.1	3.1324407784422594	0.313244077844226
100	重新	1	10	397	7521	0.1	2.941518106139213	0.2941518106139213
100	错	1	10	1000	7521	0.1	2.017699107844266	0.2017699107844266
1000	了	1	12	4817	7521	0.0833333...	0.44554778019618224	0.037128981683015184
1000	出现	1	12	104	7521	0.0833333...	4.281063487685031	0.3567552906404192
1000	和	1	12	250	7521	0.0833333...	3.4039934689641567	0.2836661224136797
1000	就	1	12	621	7521	0.0833333...	2.4941233048929243	0.20784360874107702
1000	情况	1	12	41	7521	0.0833333...	5.211882320122095	0.43432352667684127
1000	次	1	12	174	7521	0.0833333...	3.766399087611874	0.3138665906343228
1000	用	1	12	1119	7521	0.0833333...	1.9052636785144779	0.1587719732095398

图 9-15　“TF-IDF”组件的输出结果

在所显示的数据中，我们可以看到单词“拍”，所在的文档共 10 个词，而该词出现 2 次，则其 TF 值为 0.2。在本节后面的计算中，主要使用单词的 TF 值。

第五步：三元组转 kv 

该操作主要是进行格式变换。将文档中所有出现的单词汇在一起，为每个单词做索引，每个单词作为一个向量的维度，索引号即为维度的编号。具体的参数设置如图 9-16 所示，需要指

定文档 ID 所在的 ID 列；将 word 列作为向量的维度，即 key；该维度对应的数值，选择单词在文档中对应的 TF 值；由于训练数据进行此操作时，是重新生成索引，没有输入索引表，所以与此相关的两个参数选项空缺；另外，对于分隔符的设置，这里使用了默认设置。

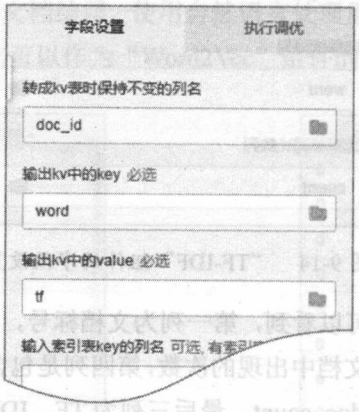


图 9-16 “三元组转 kv”组件的字段设置

执行此操作会产生两个输出。我们先查看输出桩 1，它是转换为 kv 格式的结果，将每个单词作为一个向量的维度，索引值即向量分量的位置，每个文档作为一条记录，用文档 ID 标识，该文档中的单词索引和 TF 值，构成了 kv 格式的稀疏向量表示，如图 9-17 所示。

id	key_value
1436	38:0.05263157894736842,371:0.05263157894736842,826:0.05263157894736842,842:0.05263157894736842...
2272	714:0.09090909090909091,842:0.09090909090909091,895:0.09090909090909091,1060:0.090909090909090...
3146	670:0.08333333333333333,826:0.08333333333333333,842:0.1666666666666667,2223:0.0833333333333333...
4000	826:0.1428571428571428,1308:0.1428571428571428,2082:0.1428571428571428,3268:0.1428571428571428...
4858	645:0.01886792452830189,759:0.01886792452830189,818:0.01886792452830189,826:0.0188679245283018...
5730	842:0.25,1997:0.125,2753:0.125,2896:0.125,3036:0.125,5175:0.125,5212:0.125
6596	669:0.1666666666666667,842:0.1666666666666667,1898:0.1666666666666667,3872:0.1666666666666667,4...
7466	842:0.1111111111111111,2178:0.1111111111111111,2525:0.1111111111111111,2896:0.1111111111111111,2984:0.11...
8340	804:0.125,842:0.25,3872:0.125,3889:0.125,3982:0.125,4969:0.125,5052:0.125
9215	594:0.07142857142857142,736:0.07142857142857142,1457:0.07142857142857142,1865:0.07142857142857...
1140	526:0.04545454545454546,567:0.09090909090909091,759:0.04545454545454546,818:0.045454545454454...
200	2044:0.1666666666666667,2404:0.1666666666666667,4780:0.1666666666666667,4794:0.166666666666666...
2858	621:0.07142857142857142,779:0.07142857142857142,914:0.07142857142857142,1319:0.07142857142857...
3722	522:0.06666666666666667,645:0.06666666666666667,670:0.06666666666666667,759:0.066666666666666...

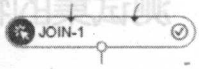
图 9-17 “三元组转 kv”组件输出桩 1 的输出结果

我们再查看输出桩 2，输出 key 的索引表，如图 9-18 所示。测试集需要使用此输出表，使

其特征索引与训练集一致。

key	key_id
/	1
0	2
0.1	3
0.3	4
0.33	5
00	6
00000000000000000	7
006	8
0096	9
00 : 00 : 27	10
02	11
02081311683	12
02日	13
03	14

图 9-18　“三元组转 kv”组件输出桩 2 的输出结果

第六步，是将产生的 kv 格式稀疏特征与原先的分类标签对应起来 

需要使用数据“JOIN”组件。该组件的输入，一个是词频统计前的结果，一个是生成稀疏特征后的结果，其中的文档 ID 是不变的，可以将此作为关联条件；设置连接类型为内连接；并选择相应的内容列、是否赔付的标签列、生成的特征列，如图 9-19 所示。

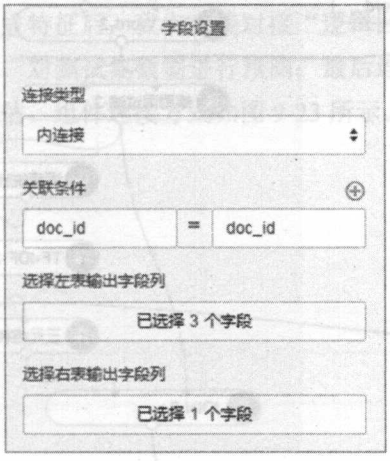


图 9-19　“JOIN”组件的参数设置

产生的结果如图 9-20 所示，各输出字段依次为 doc_id、label、content 和 key_value。

doc_id	label	content	key_value
0	1	因为不适合这款宝贝过敏	645:0.1428571428571428,1955:0.1428571428571428,2279:0.1428571428571428,341...
3	0	已经说好了可以直接退款了	842:0.25,1699:0.125,2178:0.125,2450:0.125,4007:0.125,4770:0.125,5052:0.125
6	0	因为朋友说她有此产品要送给我	895:0.1,1955:0.1,2176:0.1,2753:0.1,3234:0.1,3253:0.1,3431:0.1,4663:0.1,4770:0.1,5059...
9	1	我女朋友用了各种过敏今天去皮肤...	592:0.025,621:0.025,758:0.025,842:0.125,931:0.025,1475:0.025,1479:0.025,1595:0.025...
12	1	用了以后脸上发出了一层痘痘过敏...	522:0.09090909090909091,842:0.2727272727272727,980:0.09090909090909091,162...
18	0	要出差了收不到货不想要了	645:0.2,842:0.2,1327:0.1,1385:0.1,2705:0.1,3023:0.1,4663:0.2
21	0	跟您联系过的啥上班记得给我处...	524:0.07692307692307693,611:0.07692307692307693,1858:0.07692307692307693,1...
24	0	物流显示已签收但未收到货不知道...	700:0.07142857142857142,842:0.07142857142857142,1029:0.07142857142857142,1...
33	0	没有收到替换装很生气	2606:0.2,3025:0.2,3223:0.2,3510:0.2,3869:0.2
36	1	用了之后脸脸上起了痘痘还很红...	804:0.07692307692307693,842:0.2307692307692308,2606:0.07692307692307693,38...
39	1	第一天用就过敏脸部出现红斑眼睛...	1330:0.06666666666666667,1625:0.06666666666666667,1637:0.06666666666666666...
42	0	别人帮我买了店主不好意思	670:0.1428571428571428,826:0.1428571428571428,842:0.1428571428571428,1374:...
45	0	啊啊哒改个地址都改不了那我买...	645:0.07142857142857142,759:0.1428571428571428,826:0.07142857142857142,842...
51	0	不想要了别人前几天就已经发货了	645:0.08333333333333333,842:0.1666666666666667,1312:0.08333333333333333,13...

图 9-20 “JOIN”组件的输出结果

9.3 测试集的特征生成

生成预测数据的特征整体流程如图 9-21 所示，基本流程与“生成训练数据的特征”一致，都使用同一个停用词表“common_stopwords”。

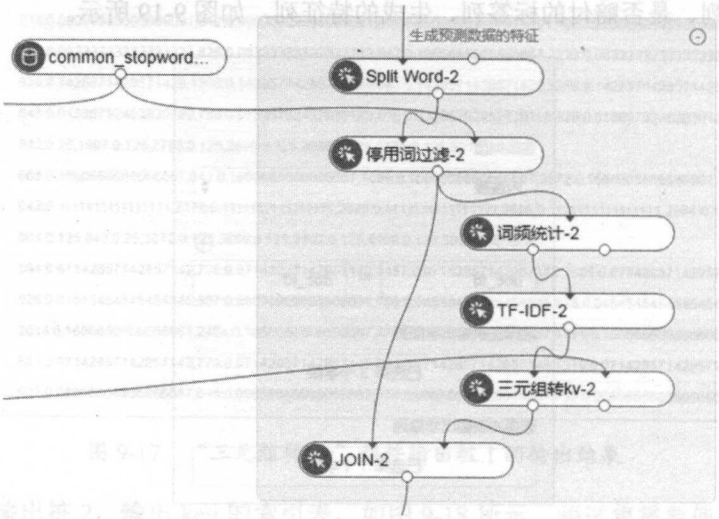


图 9-21 生成预测数据的特征

需要注意的是，“三元组转 kv”组件 三元组转kv-2 需要输入 key 的索引表，使其特征索引与训练集一致。具体的参数设置如图 9-22 所示，需要指定文档 ID 所在的 ID 列；将 word 列作为向量的维度，即 key；该维度对应的数值，选择单词在文档中对应的 TF 值；输入索引表 key 的列名为“key”，输入索引表 key 索引号的列名为“key_id”；另外，对于分隔符的设置与训练特征生成的对应组件相同，使用了默认设置。

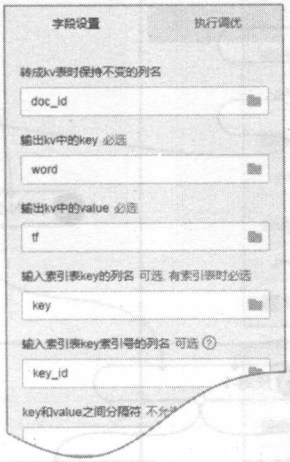


图 9-22　“三元组转 kv”组件的参数设置

9.4　模型训练、预测、评估

对训练数据和测试数据生成特征后，我们直接对接“逻辑回归二分类”组件，生成二分类模型；然后使用“预测”组件，对测试集数据进行预测；最后连接“二分类评估”和“混淆矩阵”组件，对模型效果进行评估。组件连接方式如图 9-23 所示。

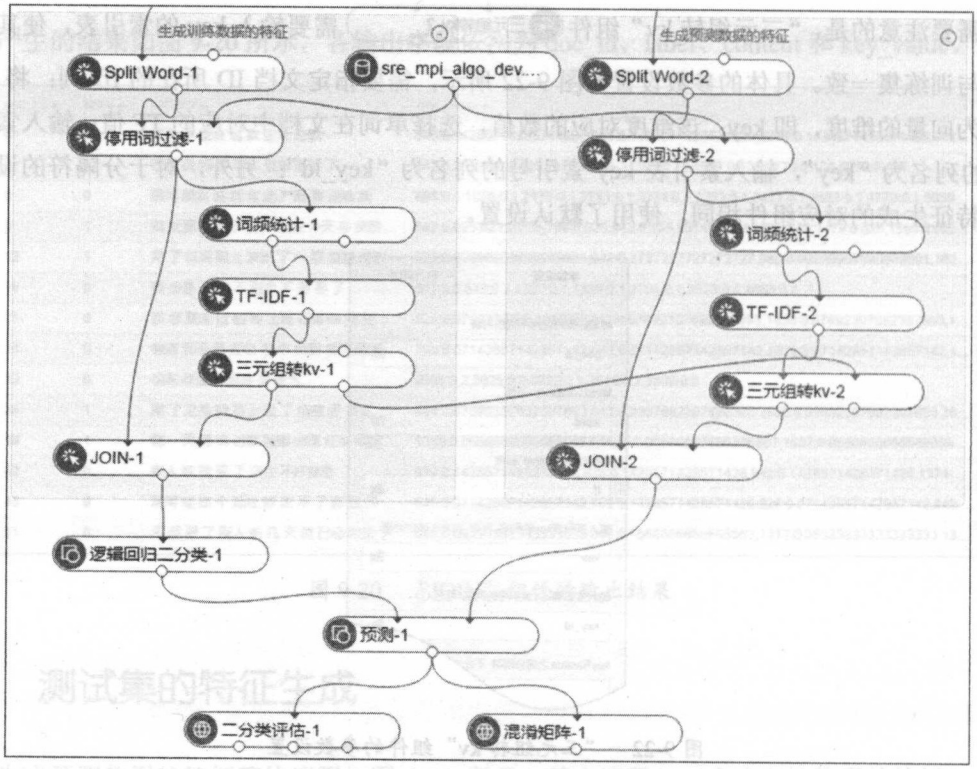


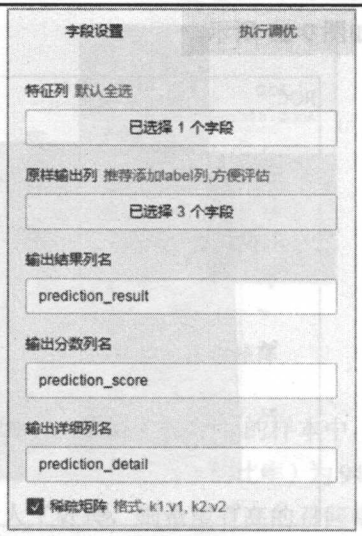
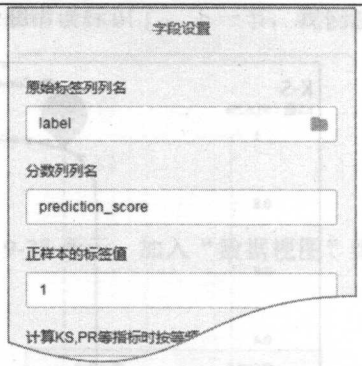
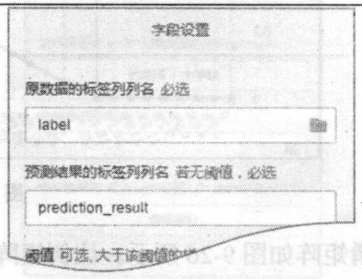
图 9-23　模型训练、预测、评估流程

组件的具体设置如表 9-2 所示。

表 9-2　各组件参数设置

逻辑回归二分类-1 选择特征列为 key_value 字段； 设定目标列为 label 字段； 勾选最下方的“是否稀疏数据”	字段设置　参数设置　执行调优 训练特征列 必选 支持Double/Int类型字段 已选择 1 个字段 目标列 必选 label 正类值 必选 eg. 0/1分类中1是正类 1 是否稀疏数据 k:v,k:v 类型特征

续表

预测-1 选择特征列为 key_value 字段； 原样输出列选择 label、content、doc_id 字段； 勾选最下方的“是否稀疏数据”	
二分类评估-1 设置原始标签列列名为 label； 正样本的标签值为 1	
混淆矩阵-1 设置原始标签列列名为 label	

我们查看最后的评估结果，模型的 ROC 评估曲线如图 9-24 所示，AUC 值为 0.98；K-S 评估曲线如图 9-25 所示。

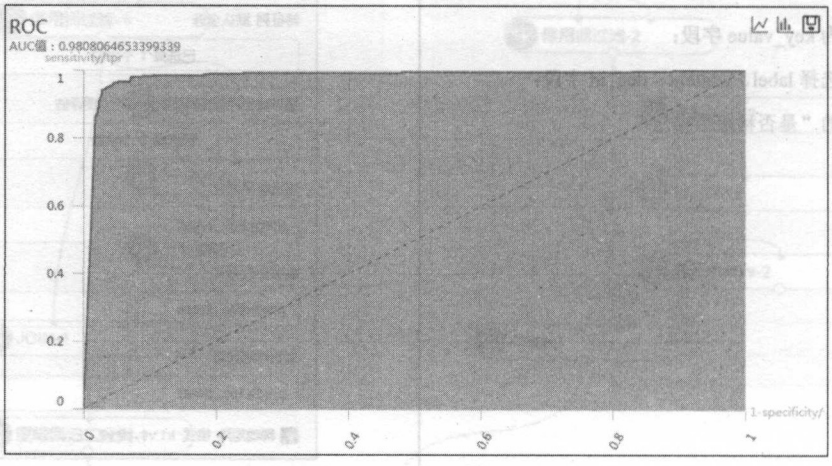


图 9-24　ROC 评估曲线

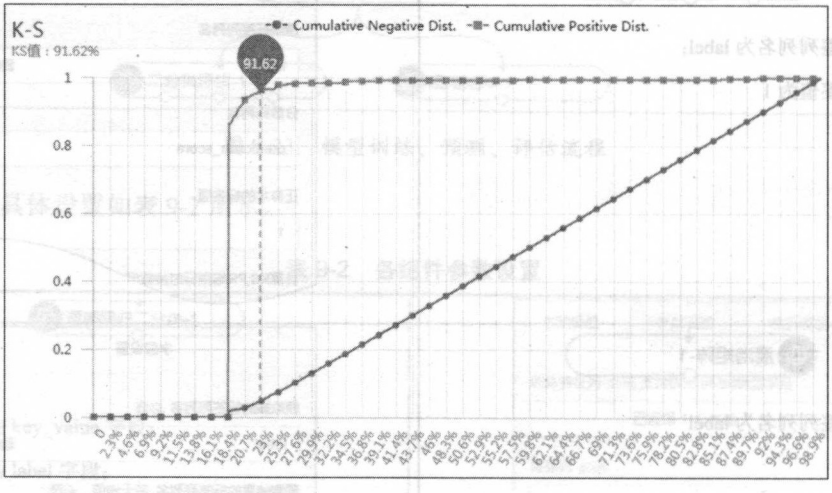


图 9-25　K-S 评估曲线

混淆矩阵如图 9-26 所示；比例矩阵如图 9-27 所示。

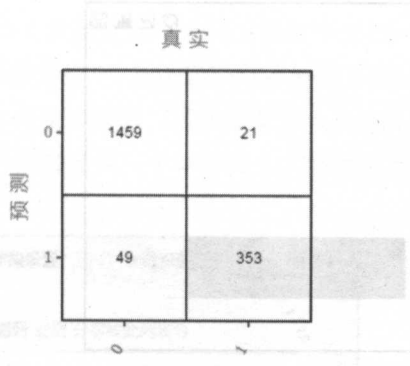


图 9-26　混淆矩阵

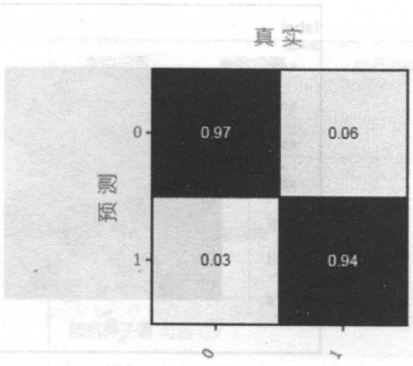


图 9-27　比例矩阵

我们关注标签为 1，即赔付的责任相关案件，其准确率（预测标签为 1 的样本中，真实标签为 1 的比率）为 87.811%，召回率（真实标签为 1 的样本中，预测标签为 1 的比率）为 94.385%。如果我们要将预测结果用来初选，缩小范围，然后进行人工处理，则希望有高的召回率，即尽量标识出需要赔付的案件，即使有些不需要赔付的案件被错误标识了。下一节，我们演示正负样本配置比例对模型结果的影响，就会对召回率有影响。

9.5　提高召回率

我们先做个小实验，在训练数据生成特征后，如图 9-28 所示，加入“数据视图”组件，并选择特征列为 label 字段，如图 9-29 所示。

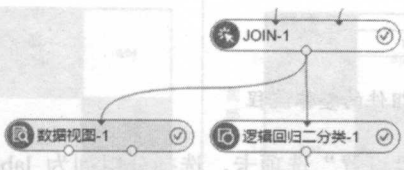


图 9-28　连接“数据视图”组件

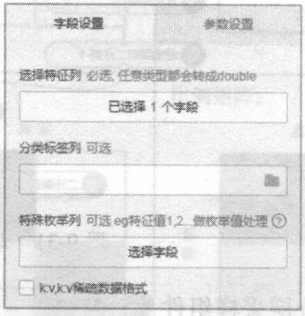


图 9-29　“数据视图”组件的字段设置

运行结果显示如图 9-30 所示，可知 label=0 的样本数约为 6000，label=1 的样本数约为 1500，二者的比例接近 4∶1。

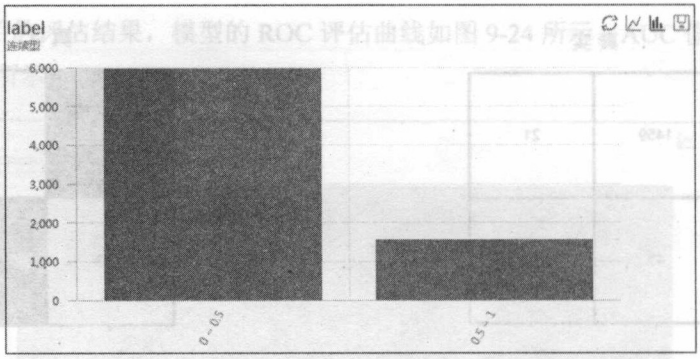


图 9-30　“数据视图”组件的运行结果

我们将通过加入“分层采样”组件来改变参加模型训练的各类样本的数量，加入该组件后，整体的流程如图 9-31 所示。

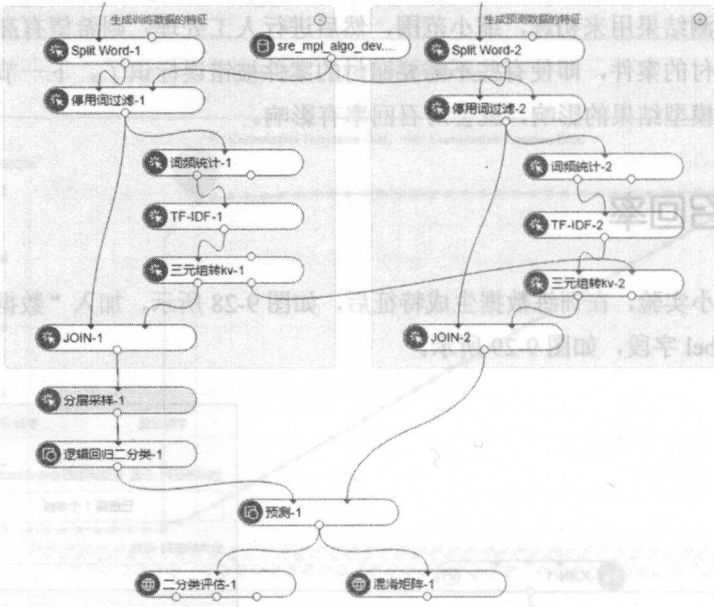


图 9-31　加入“分层采样”组件的整体流程

设置分层采样组件 ，在“字段设置”选项卡，选择分组列为 label，如图 9-32 所示；在“参数设置”选项卡的设置如图 9-33 所示，采样比例可以对不同的标签值分别设置，譬如：“0：0.5, 1：1.0”就表示 label=0 的样本采样 50%，label=1 的样本采样 100%，为了实验可重复，设置随机种子为固定值，这里设为 1234567。

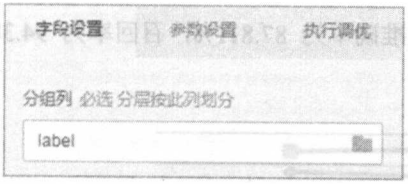


图 9-32　分层采样组件的字段设置

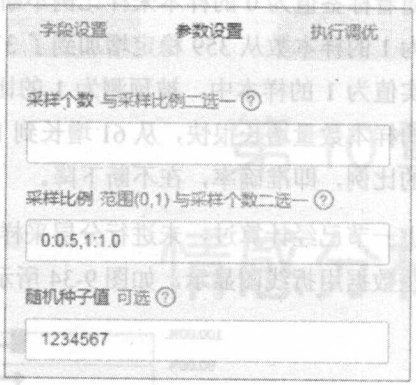
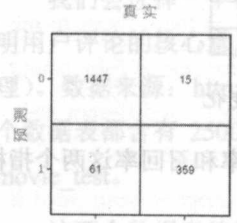
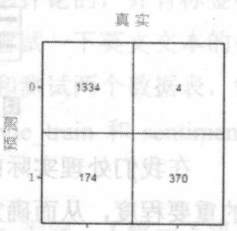
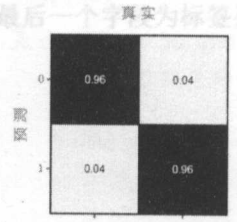
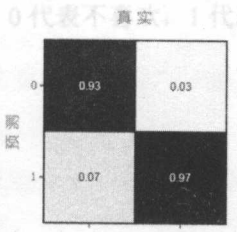
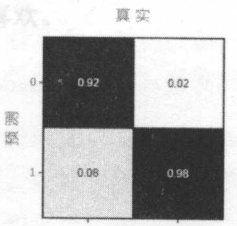
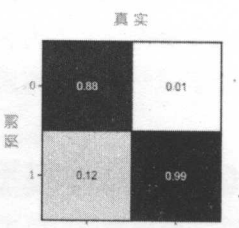


图 9-33　分层采样组件的参数设置

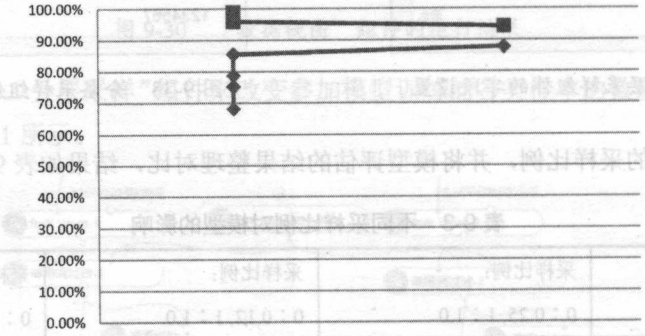
变化分层采样的采样比例，并将模型评估的结果整理对比，结果如表 9-3 所示。

表 9-3　不同采样比例对模型的影响

采样比例： 0∶0.5, 1∶1.0	采样比例： 0∶0.25, 1∶1.0	采样比例： 0∶0.12, 1∶1.0	采样比例： 0∶0.06, 1∶1.0
混淆矩阵： 	混淆矩阵： 	混淆矩阵： 	混淆矩阵： 
比例矩阵： 	比例矩阵： 	比例矩阵： 	比例矩阵： 
准确率：85.476% 召回率：95.989%	准确率：78.618% 召回率：97.326%	准确率：75.102% 召回率：98.396%	准确率：68.015% 召回率：98.93%

随着标签值为 0 的样本采样比例不断减小，可以看到混淆矩阵中，真实值为 1 的样本中，预测为 1 的样本数从 359 稳定增加到了 370；预测为 0 的样本数相应地从 15 逐渐降到了 4，所以真实值为 1 的样本中，被预测为 1 的比率，即召回率，在不断提高。而真实值为 0，被预测为 1 的样本数量增长很快，从 61 增长到 174，从而使预测为 1 的样本中，真实值也为 1 的样本所占的比例，即准确率，在不断下降。

上一节已经计算过，未进行分层采样前的数据的准确率为 87.811%，召回率为 94.385%，将这些数据用折线图显示，如图 9-34 所示。



	1	0.5	0.25	0.12	0.06
准确率	87.81%	85.48%	78.62%	75.10%	68.02%
召回率	94.39%	95.99%	97.33%	98.40%	98.93%

图 9-34　准确率和召回率随标签值为 0 的样本采样比例的变化

在我们处理实际问题时，需要根据所解决问题的目标，来确定准确率和召回率这两个指标的重要程度，从而确定一个合理的不同标签样本的配比。

第10章 情感分析

本章我们介绍一个文本分析领域非常有用的应用——情感分析。我们在网上购买商品，浏览电影信息，寻找饭馆信息时，经常看到一个区域显示着用户的评论，我们习惯性地参考其他人的评论，如果大家都说好，就更坚定了我们买下此商品的信心，如果有人给了差评，我们会很自然地想到，我要是购买了是否也会遇到同样的问题。那么，如何通过机器学习算法，高效地判断出评论的情感倾向呢？

我们会选择一个在情感分析方面常用的公开数据集，是用户关于电影评论的，并有标签标明用户评论的核心意思是喜欢还是厌恶，内容是英文的（☺ 我们正好尝试一下英文文本的处理）。数据来源：http://ai.stanford.edu/~amaas/data/sentiment/ ，包括训练和测试两个数据表，每个数据表都含有 25000 条电影评论，数据表的名称分别为 sentiment_movie_train 和 sentiment_movie_test。

这两个数据表的结果相同，我们以训练数据表 sentiment_movie_train 为例，了解一下将要分析的数据情况，如图 10-1 所示，第一个字段是评论的 ID，第二个字段是评论的具体内容，最后一个字段为标签列，0 代表不喜欢，1 代表喜欢。

review_id	review	label
100000	Oh dear. good cast, but to write and direct is an art and to write wit and direct wit is a bit of a task. Even doing good comedy you have ...	0
100001	This movie is a perfect adaptation of the English Flick Unfaithful. Ashmit plays the role of Richard Gere, Emran that of Olivier and Mali...	0
100002	I recently purchassed the very underrated Dreamcast and went off into town to find some games to use them on. I bought Soul Calibu...	1
100003	If you are a traveller, if there is a fire burning into your heart, if you'd call "home" every place on earth, but none of them can give you e...	1
100004	This film exhibits artful cinematic techniques wherein instead of landscape capturing the attention of the camera it is small details in h...	1
100005	I've seen a lot of movies and rarely would I ever rate a movie "1" but this movie was beyond terrible. The acting was terribl...	0
100006	Drawing Restraint 9. dir: Matthew Barney. How do you know when you're in the middle of a pretentious art film? Is it that th...	0
100007	This movie really kicked some ass. I watched it over and over and it never got boring. Angelina Jolie really kicked some ass in the mo...	1
100008	Hallam Foe tells us the story about a boy who lost his mother and experiences some sort of Oedepus complex afterward. I...	0
100009	Typical De Palma movie made with lot's of style and some scene's that will bring you to the edge of your seat. Most certain...	1
100010	This is a musical adaptation of Dicken's "Oliver Twist". For the most part, the original story has been maintained, though for the flow of...	1
100011	This movie was a long build-up with no climax. People whom refer to the swordfight in the end as great must either be out of their min...	0

图 10-1　数据表 sentiment_movie_train 的内容

每条评论都比较长，这里将第一条的完整内容显示如下，虽然有“good”和“laughed loud”等好评的词，但整篇读下来，我们知道用户并不看好这部电影。从这里可以感受到情感分析的难度。

> Oh dear. good cast, but to write and direct is an art and to write wit and direct wit is a bit of a task. Even doing good comedy you have to get the timing and moment right. Im not putting it all down there were parts where i laughed loud but that was at very few times. The main focus to me was on the fast free flowing dialogue, that made some people in the film annoying. It may sound great while reading the script in your head but getting that out and to the camera is a different task. And the hand held camera work does give energy to few parts of the film. Overall direction was good but the script was not all that to me, but I'm sure you was reading the script in your head it would sound good. Sorry.

本章所要解决的问题是判断一段评论表达的是喜欢还是不喜欢，这还是二分类预测问题，重点是如何从文本内容中构造出特征。本章除了使用前一章所用的词袋（bag of words）方式，还会尝试一种新的方式：词向量（word embedding）。

10.1　词袋模型

借鉴上一章在对文本的处理时使用的“词袋”方式，将每个词作为一个特征，而该词在样本中的词频（Term Frequency）值作为该特征的属性值，每个样本只包含部分的单词，其对应值非 0，通常会将特征用 kv 的稀疏格式进行存储。然后使用逻辑回归二分类组件对构造出来的高

维训练样本进行训练，并对测试数据进行预测，评估模型效果。

整体流程如图 10-2 所示，包括三大部分：生成训练数据的特征、生成测试数据的特征、模型训练预测评估。

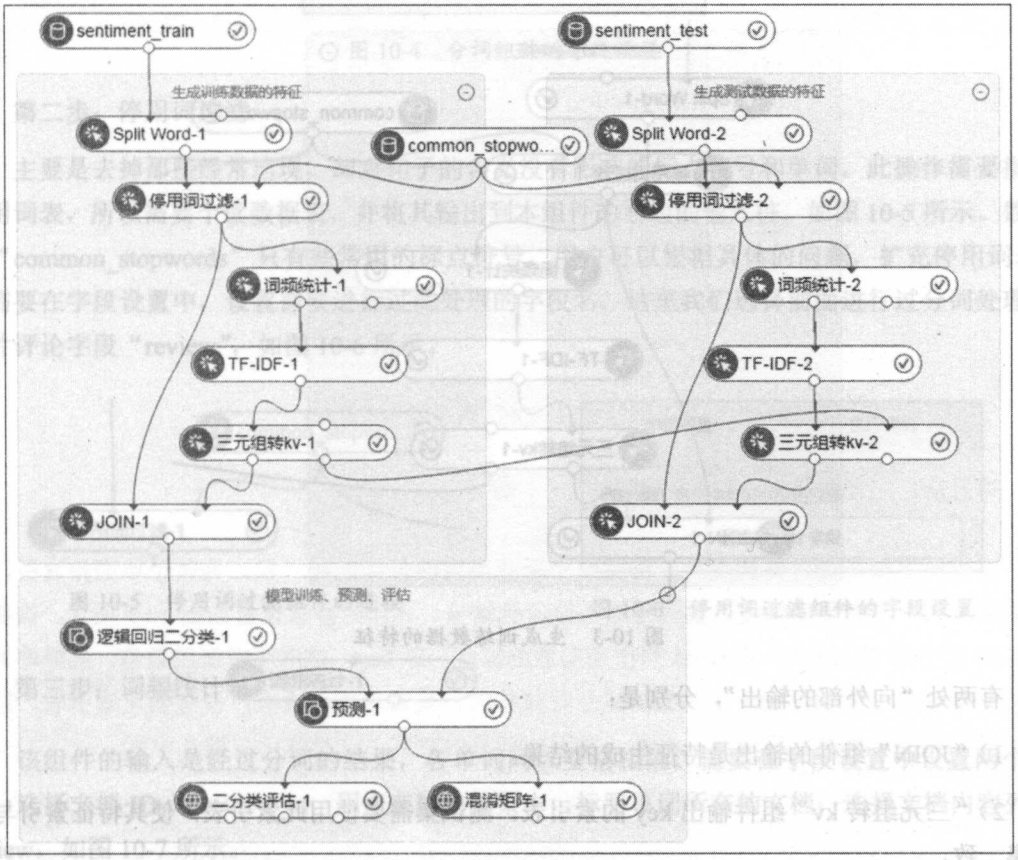


图 10-2 整体流程图

我们需要关注两个重要的功能区“生成训练数据的特征”、“生成预测数据的特征”。

- 相同点：基本流程一致；都使用同一个停用词表“common_stopwords”。
- 差异点：在“三元组转 kv”组件的使用上，根据训练数据会产生“key 的索引表”，即组件的第二个输出桩；为了保证预测产生的特征索引的一致性，需要通过组件的第二个输入桩，引用该“key 的索引表”。

10.1.1 训练集的特征生成

整体流程如图 10-3 所示，共有六个处理组件；除了原始的语料标记数据，还需要使用停用词表“common_stopwords”。

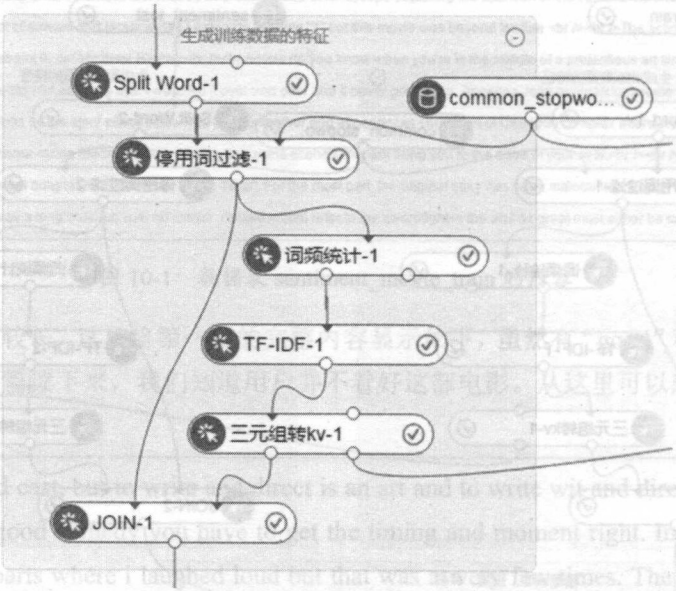


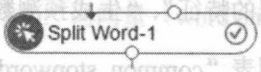
图 10-3 生成训练数据的特征

有两处“向外部的输出”，分别是：

1）“JOIN”组件的输出是特征生成的结果。

2）“三元组转 kv”组件输出 key 的索引表。测试集需要使用此索引表，使其特征索引与训练集一致。

下面会对每一步操作进行详细介绍。

第一步：分词 

需要在“字段设置”选项卡中，设置需要分词处理的字段名，这里我们选择评论字段“review”，其余的参数使用默认设置，如图 10-4 所示。这里需要说明一点，分词组件在这里是大材小用了。英文单词本身就是以空格作为分割的，不像中文中需要确定哪些汉字是属于一个单词，分词模块最核心的功能对英文文本来说，是没有作用的，但是英文中的标点符号与单词间没有空格，所以在这里使用分词组件达到此目的。

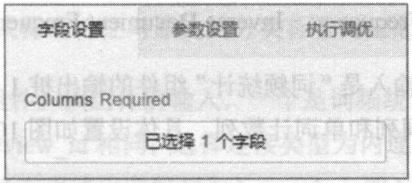


图 10-4　分词组件的字段设置

第二步：停用词过滤

主要是去掉那些经常出现，却对句子的含义没有影响的标点符号和单词。此操作需要指定停用词表，所以需要个读数据表，并将其输出到本组件的右边的输入桩，如图 10-5 所示。数据表“common_stopwords”只有些常用的标点符号，用户可以根据具体的问题，扩充停用词表。还需要在字段设置中，设置需要进行过滤处理的字段名，这里我们选择前面进行过分词处理的影片评论字段“review”，如图 10-6 所示。

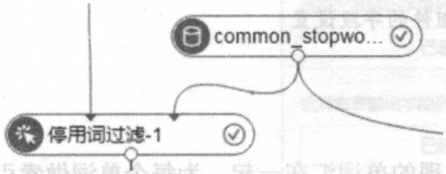


图 10-5　停用词过滤组件的连接

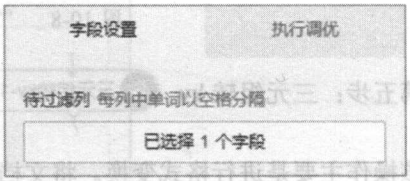
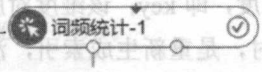


图 10-6　停用词过滤组件的字段设置

第三步：词频统计



该组件的输入是经过分词的结果，各单词间以空格相隔。需要在字段设置中设置两个参数，选择文档 ID 为 review_id，用来在输出结果中，标明单词所在的文档；选择文档内容列为 review，如图 10-7 所示。

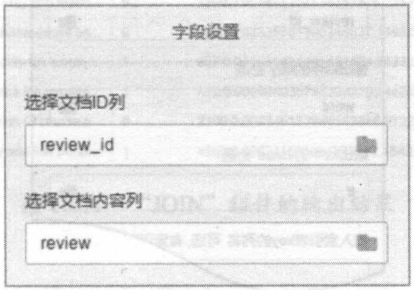
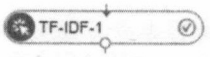


图 10-7　词频统计组件的字段设置

第四步：TF-IDF（Term Frequency - Inverse Document Frequency） 

对于“TF-IDF”组件，其输入是“词频统计”组件的输出桩 1 的结果，需要在字段设置中，配置三个参数：文档 ID，单词列和单词计数列。具体设置如图 10-8 所示。

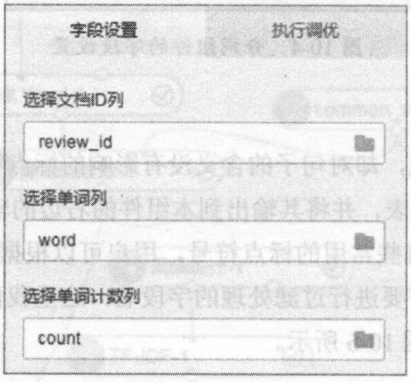
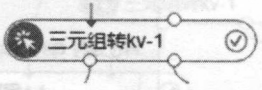


图 10-8 “TF-IDF”组件的字段设置

第五步：三元组转 kv 

该操作主要是进行格式变换。将文档中所有出现的单词汇在一起，为每个单词做索引，每个单词作为一个向量的维度，索引号即为维度的编号。具体的参数设置如图 10-9 所示，需要指定文档 ID 所在的 ID 列；将 word 列作为向量的维度，即 key；该维度对应的数值，选择单词在文档中对应的 TF 值；由于训练数据进行此操作时，是重新生成索引，没有输入索引表，所以与此相关的两个参数选项空缺；另外，对于分隔符的设置，这里使用了默认设置。

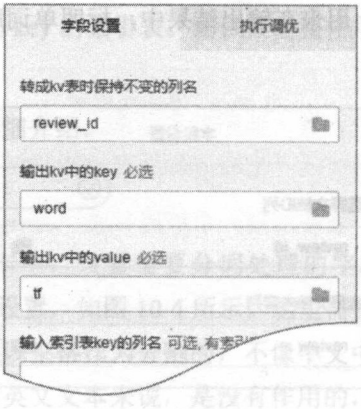
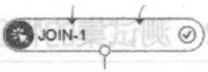


图 10-9 “三元组转 kv”组件的字段设置

第六步：将产生的 kv 格式稀疏特征与原先的分类标签对应起来 

需要使用数据“JOIN”组件。该组件的输入，一个是词频统计前的结果，一个是生成稀疏特征后的结果，关联条件为 review_id 相同；选择连接类型为内连接，并选择左表输出的字段为 review_id，review 和 label；选择右表输出字段列为 key_value，即 kv 格式稀疏特征列，如图 10-10 所示。

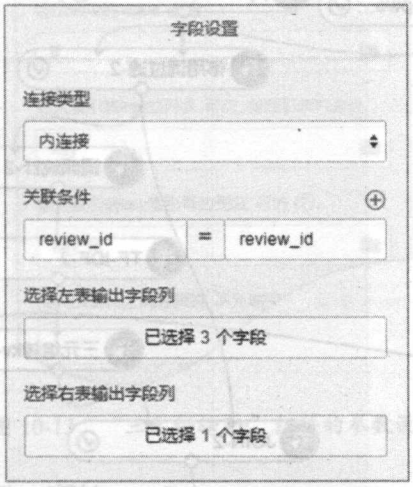


图 10-10　“JOIN”组件的参数设置

产生的结果如图 10-11 所示。

review_id	review	label	key_value
100224	I know it's crude and I know that it isn t at all P...	1	895:0.008130081300813009,10613:0.008130081300813009,16870:0.0081300...
100992	Alright before we review I have to ask why isn...	1	573:0.002309468822170901,987:0.002309468822170901,2239:0.0023094688...
101760	br br What an absolutely crappy film this is Ho...	0	5221:0.01754385964912281,5586:0.01754385964912281,8233:0.0175438596...
102528	This movie had good intentions and a good s...	0	813:0.0053763440860215 06,926:0.0053763440860215 06,7614:0.0053763440...
103296	Weak Bobby Pineapple Salsa Flay and Mario...	0	2269:0.007575757575757576,2357:0.007575757575757576,2360:0.00757575...
104064	Working with one of the best Shakespeare so...	0	6037:0.02325581395348837,14982:0.02325581395348837,36671:0.02325581...
104832	This is Paul F Ryan s first and only full length ...	1	987:0.0014814814814814 81,2123:0.011851851851851 85,2445:0.0014814814...
105600	Perspective is a good thing Since the release...	1	413:0.0009940357852882703,440:0.0009940357852882703,451:0.000994035...
106368	First of all excuse my bad English Of course o...	0	2360:0.003846153846153846,9722:0.003846153846153846,10631:0.0038461...
107136	For a film that got little publicity and few peopl...	1	4500:0.006211180124223602,4643:0.006211180124223602,4857:0.00621118...

图 10-11　“JOIN”组件的输出结果

10.1.2 测试集的特征生成

整体流程如图 10-12 所示，基本流程与“生成训练数据的特征”一致，都使用同一个停用词表“common_stopwords”。

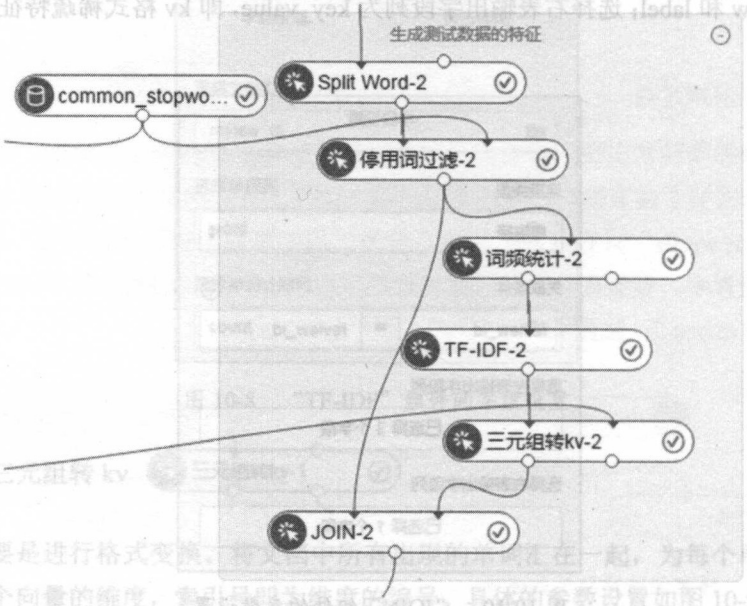


图 10-12 生成测试数据的特征

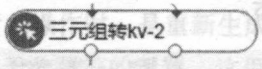
需要注意的是，“三元组转 kv”组件  需要输入 key 的索引表，使其特征索引与训练集一致。具体的参数设置如图 10-13 所示，需要指定文档 ID 所在的 ID 列；将 word 列作为向量的维度，即 key；该维度对应的数值，选择单词在文档中对应的 TF 值；输入索引表 key 的列名为“key”，输入索引表 key 索引号的列名为“key_id”；另外，对于分隔符的设置与训练特征生成的对应组件相同，使用了默认设置。



图 10-13 “三元组转 kv”组件的参数设置

10.1.3 模型训练、预测、评估

对训练数据和测试数据生成特征后，我们直接对接“逻辑回归二分类”组件，生成二分类模型；然后使用“预测”组件，对测试集数据进行预测；最后连接“二分类评估”和“混淆矩阵”组件，对模型效果进行评估。组件连接方式如图 10-14 所示。

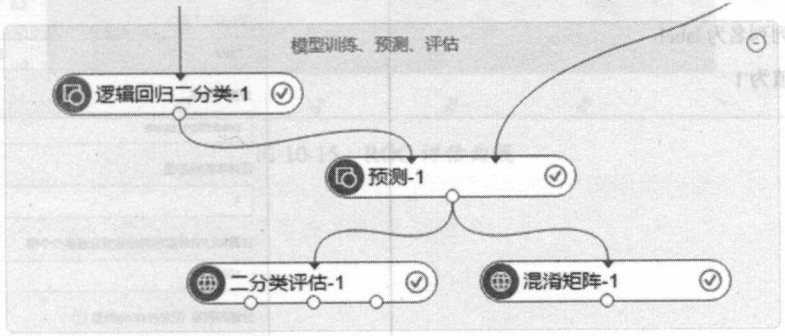
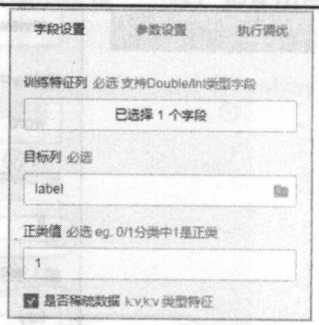
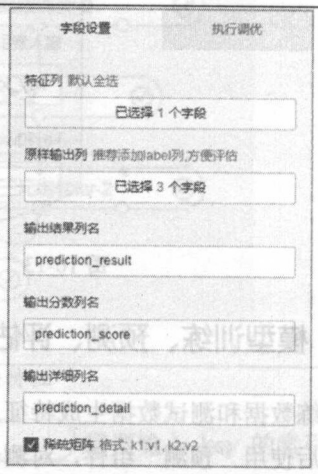


图 10-14 模型训练、预测、评估

组件的具体设置如表 10-1 所示。

表 10-1　各组件的参数设置

逻辑回归二分类-1 选择特征列为 key_value 字段， 设定目标列为 label 字段	
预测-1 选择特征列为 key_value 字段， 原样输出列选择 label、review_id、review 字段	
二分类评估-1 设置原始标签列列名为 label， 正样本的标签值为 1	

续表

混淆矩阵-1 设置原始标签列列名为 label	字段设置 原数据的标签列列名 必选 label 预测结果的标签列列名 若无阈值，必选 prediction_result 阈值 可选, 大于该阈值的样本作为正样本 预测结果的详细列列名 若指定阈值, 必选 与标签列列名不能共存 正样本的标签值 若无阈值, 可选; 否则, 必选 二分类时，指定训练系数针对的label值

我们查看最后的评估结果，模型的 ROC 评估曲线如图 10-15 所示，AUC 值为 0.9469。

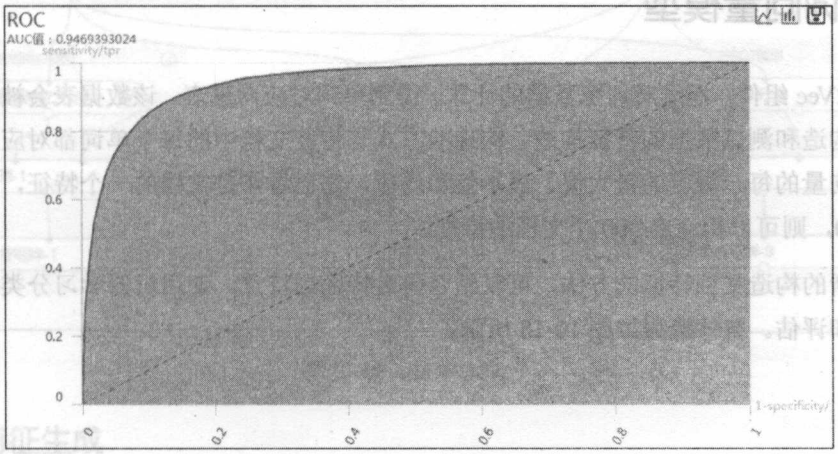


图 10-15　ROC 评估曲线

混淆矩阵如图 10-16 所示，比例矩阵如图 10-17 所示。

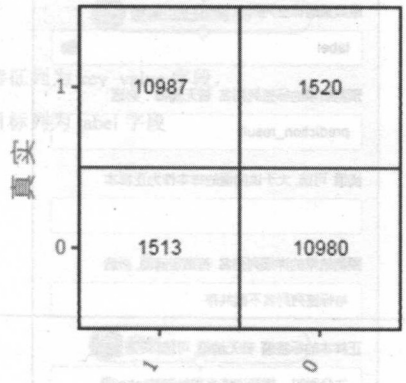


图 10-16　混淆矩阵

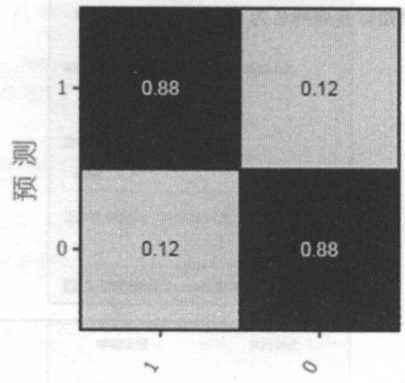


图 10-17　比例矩阵

10.2　词向量模型

Word2Vec 组件，通过对训练数据的计算，得到单词对应向量表，该数据表会被用于训练数据的特征构造和测试数据的特征构造。构造的方式是将某文档中的每个单词都对应到一个向量值，则该向量的每一维度的最大值、最小值和均值，都已看作是文档的一个特征，如果向量的维度是 100，则可以构造出 300 个文档的特征。

有了新的构造文档特征的方法，可以结合构造特征的特点，使用机器学习分类模型进行训练、预测和评估。整体流程如图 10-18 所示。

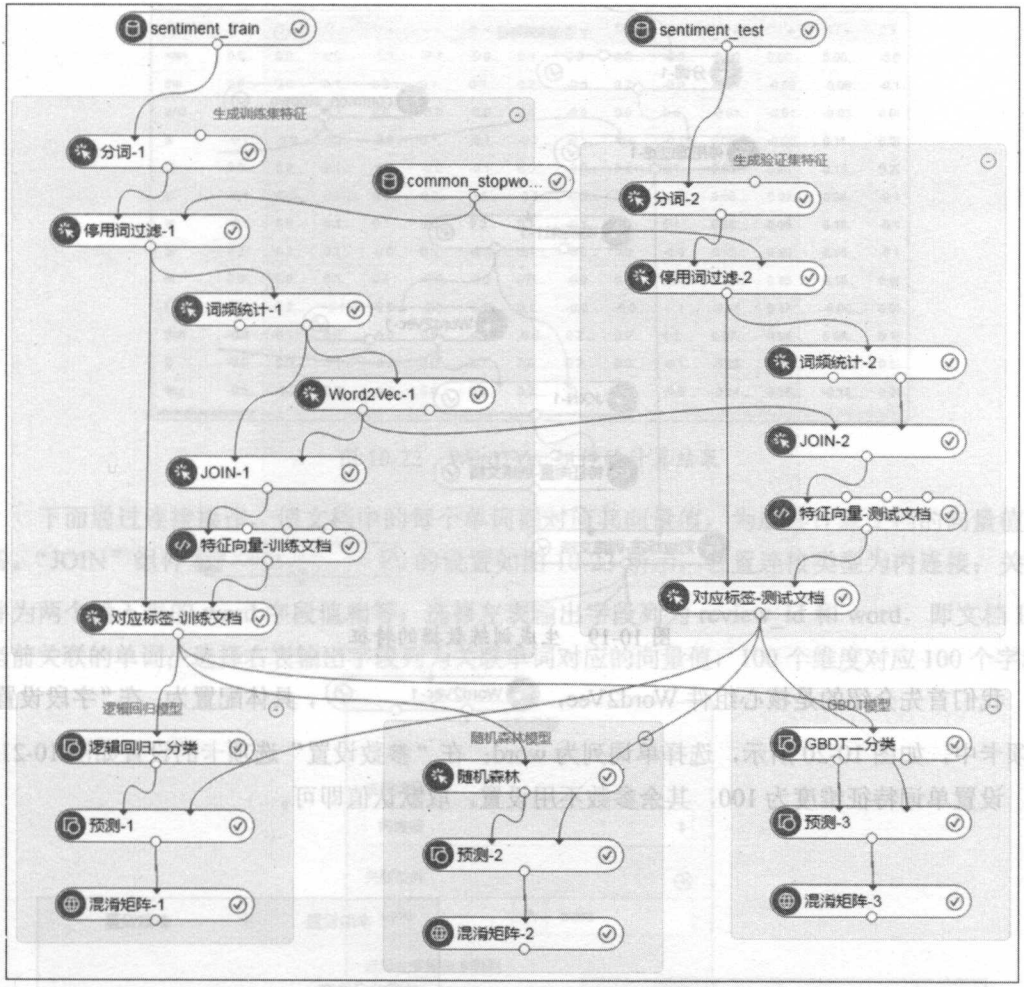


图 10-18　整体流程

10.2.1　特征生成

生成特征整体的流程如图 10-19 所示。其中，“分词”组件、“停用词过滤”组件、“词频统计”组件的设置与前面的流程相同，这里就不再重复叙述。

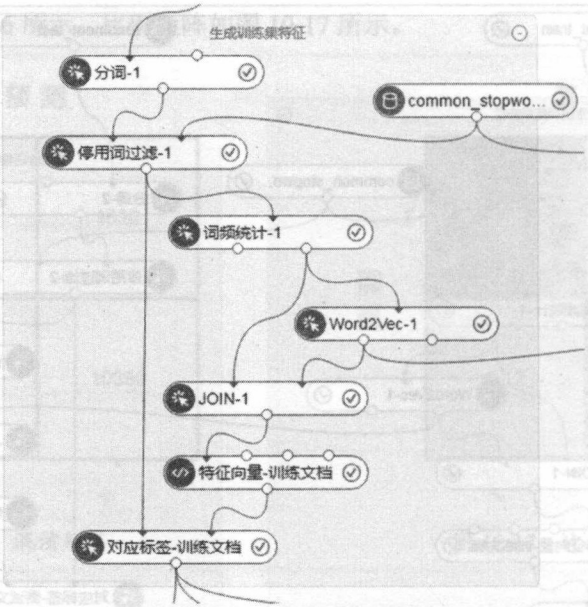
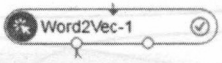


图 10-19　生成训练数据的特征

我们首先介绍的是核心组件 Word2Vec，，具体配置为：在“字段设置”选项卡中，如图 10-20 所示，选择单词列为 word；在“参数设置”选项卡的设置如图 10-21 所示，设置单词特征维度为 100，其余参数不用设置，取默认值即可。

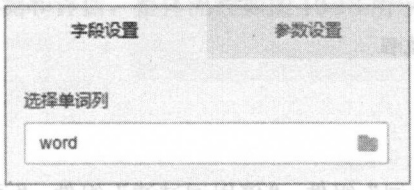


图 10-20　Word2Vec 组件的字段设置

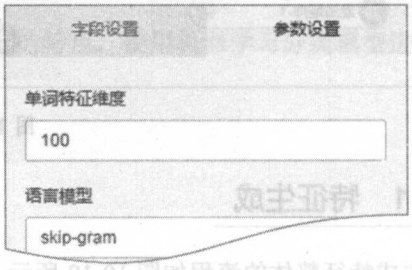


图 10-21　Word2Vec 组件的参数设置

计算结果如图 10-22 所示。

word	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13
</s>	0.0...	0.0...	0.0...	-1.6...	-7.4...	-0.0...	0.0...	0.0...	0.0...	-0.0...	-0.00...	0.00...	0.00...	-0.0
the	0.0...	0.0...	0.1...	0.0...	0.1...	0.0...	0.2...	-0.0...	0.2...	-0.0...	-0.01...	-0.03...	-0.09...	-0.1
and	-0.0...	0.0...	0.2...	0.0...	-0.0...	-0.0...	0.0...	-0.0...	0.0...	0.0...	-0.01...	-0.01...	-0.03...	0.0
a	-0.1...	-0.0...	0.3...	-0.0...	0.1...	-0.1...	-0.1...	-0.1...	-0.0...	-0.1...	0.00...	-0.00...	0.11...	0.0
of	0.0...	0.2...	0.1...	-0.2...	-0.0...	-0.1...	0.0...	-0.0...	0.1...	0.1...	0.03...	0.24...	0.15...	0.2
to	-0.1...	-0.0...	0.1...	0.0...	0.2...	0.0...	-0.1...	-0.0...	0.4...	-0.0...	0.06...	0.02...	0.04...	-0.1
is	-0.1...	0.0...	0.1...	0.1...	0.0...	0.0...	0.1...	-0.1...	0.1...	0.1...	0.00...	0.05...	0.18...	-0.1
br	0.2...	0.1...	0.2...	-0.0...	0.0...	-0.3...	0.1...	-0.0...	0.0...	-0.0...	0.12...	0.30...	0.05...	-0.1
in	0.0...	0.0...	0.1...	0.1...	-0.0...	-0.0...	0.0...	-0.0...	0.1...	0.0...	0.33...	0.35...	0.16...	0.0
I	0.0...	0.2...	-0.1...	-0.0...	-0.0...	-0.0...	0.1...	-0.0...	-0.0...	-0.1...	-0.09...	0.37...	-0.00...	0.0
that	-0.0...	0.1...	0.0...	-0.0...	0.0...	-0.0...	-0.0...	0.2...	0.0...	0.0...	0.07...	0.04...	0.06...	0.1
it	-0.0...	0.0...	0.1...	-0.2...	0.0...	0.1...	0.0...	0.2...	0.0...	-0.1...	-0.03...	0.07...	-0.24...	0.1
this	-0.0...	-0.2...	0.0...	-0.1...	0.0...	0.0...	0.2...	-0.1...	-0.0...	-0.0...	-0.14...	0.18...	-0.15...	0.2

图 10-22 Word2Vec 组件的计算结果

下面通过连接操作，使文档中的每个单词都对应其向量值，为后续计算文档的向量值做准备。“JOIN”组件 JOIN-1 的设置如图 10-23 所示，设置连接类型为内连接；关联条件为两个输入表的 word 字段值相等；选择左表输出字段列为 review_id 和 word，即文档 ID 和当前关联的单词；选择右表输出字段列为关联单词对应的向量值，100 个维度对应 100 个字段。

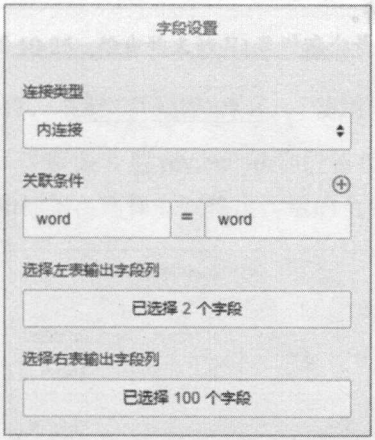


图 10-23 “JOIN”组件的参数设置

连接操作的结果如图 10-24 所示，最左边的字段为文档 ID，其次是 word 字段，然后是单词对应的 100 维向量。该组件执行的时候，没有按照输入文档单词的顺序，可以看到输出的文档 ID 没有顺序，我们看到的数据片段里，word 字段都是同一个内容：“0”，所以对应出现的 100 维向量值也是相同的。还有一点要特别说明，一个文档中的某个单词出现多次，就会被记录多次，比如本数据片段中 review_id=113669 的文档，就包含两个“0”。

review_id	word	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11
107879	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
113669	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
113669	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
109401	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
106111	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
123625	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
123433	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
103645	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
123570	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
108382	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
122503	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
100284	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...
103186	0	-0.2...	0.3...	0.1...	-0.1...	-0.1...	-0.0...	-0.2...	0.3...	0.0...	-0.4...	0.05...	-0.14...

图 10-24 “JOIN”组件的运行结果

接下来，我们就将通过文档中各个单词的向量值，构造出整个文档的特征向量值，具体方法是，对该文档中的所有单词的向量值，将每一维度的最大值、最小值和均值，都看作是文档的一个特征，如果向量的维度是 100，则可以构造出 300 个文档的特征。

特征向量-训练文档 使用 SQL 组件实现，具体的 SQL 语句如下。为节省篇幅，对于一些有规律的内容，使用了省略号代替。

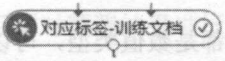
```
select review_id,
min(f0) as f1_0,
min(f1) as f1_1,
min(f2) as f1_2,
min(f3) as f1_3,
……
min(f97) as f1_97,
min(f98) as f1_98,
min(f99) as f1_99,
avg(f0) as f2_0,
avg(f1) as f2_1,
avg(f2) as f2_2,
avg(f3) as f2_3,
……
avg(f97) as f2_97,
avg(f98) as f2_98,
avg(f99) as f2_99,
max(f0) as f3_0,
max(f1) as f3_1,
max(f2) as f3_2,
max(f3) as f3_3,
```

```
……
max(f97) as f3_97,
max(f98) as f3_98,
max(f99) as f3_99
from ${t1}
group by review_id;
```

得到每个文档的特征如图 10-25 所示。

review_id	f1_0	f1_1	f1_2	f1_3	f1_4	f1_5	f1_6	f1_7	f1_8	f1_9	f1
100000	-0.329...	-0.709...	-0.463...	-0.468...	-0.614...	-0.747...	-0.600...	-0.664...	-0.383...	-0.540...	-(
100001	-0.499...	-0.600...	-0.651...	-0.669...	-0.546...	-0.379...	-0.751...	-0.550...	-0.488...	-1.238...	-(
100002	-0.503...	-0.483...	-0.790...	-0.904...	-0.453...	-0.765...	-1.034...	-0.664...	-0.430...	-0.887...	-'
100003	-0.395...	-0.375...	-0.413...	-0.547...	-0.528...	-0.438...	-0.373...	-0.664...	-0.268...	-0.447...	-(
100004	-0.557...	-0.806...	-0.810...	-0.623...	-0.940...	-0.483...	-0.510...	-0.671...	-0.690...	-0.793...	-(
100005	-0.499...	-0.589...	-0.570...	-0.605...	-0.412...	-0.597...	-0.497...	-0.854...	-0.365...	-0.760...	-(
100006	-1.284...	-1.097...	-1.115...	-0.832...	-1.249...	-1.214...	-0.916...	-0.869...	-0.954...	-0.998...	-(
100007	-0.290...	-0.516...	-0.903...	-0.846...	-0.738...	-0.339...	-0.762...	-0.421...	-0.237...	-0.942...	-(
100008	-1.284...	-0.558...	-0.927...	-0.832...	-0.622...	-0.653...	-0.724...	-0.664...	-0.654...	-0.781...	-(
100009	-1.015...	-0.569...	-0.616...	-1.259...	-0.631...	-0.553...	-0.453...	-0.664...	-0.637...	-0.649...	-(
100010	-0.537...	-0.479...	-1.443...	-0.662...	-0.518...	-0.588...	-1.163...	-0.564...	-0.839...	-0.601...	-(
100011	-0.527...	-0.352...	-0.670...	-0.749...	-0.842...	-0.543...	-0.683...	-0.564...	-0.388...	-1.253...	-(
100012	-0.633...	-0.909...	-0.930...	-0.705...	-0.635...	-0.853...	-1.071...	-1.010...	-0.425...	-0.963...	-(

图 10-25 输出的文档 ID 及对应向量值

还需要连接评论的标签字段，

，使用“JOIN”组件实现，如图 10-26 所示，设置连接类型为内连接；关联条件是 review_id 字段值相等；选择左表输出两个字段列：review_id 和 label，即评论 ID 和标签；选择右表输出构造的全部 300 个特征字段。

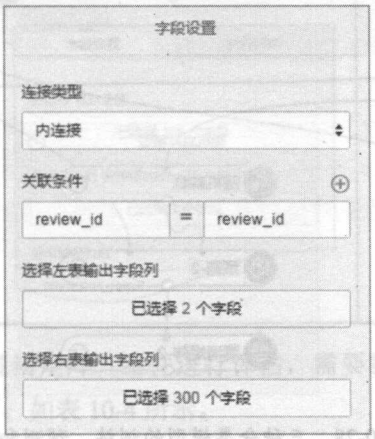


图 10-26 “JOIN”组件的参数设置

结果数据如图 10-27 所示，可以看到首先是评论 ID，然后是标签列，随后是 300 个特征字段。

review_id	label	f1_0	f1_1	f1_2	f1_3	f1_4	f1_5	f1_6	f1_7	f1_8
100224	1	-0.342...	-0.675...	-0.537...	-0.832...	-0.486...	-0.633...	-0.473...	-0.664...	-0.318...
100992	1	-0.653...	-1.255...	-0.867...	-0.908...	-1.513...	-0.866...	-1.215...	-1.156...	-0.785...
101760	0	-0.686...	-0.662...	-0.844...	-0.489...	-0.826...	-0.444...	-0.540...	-0.733...	-0.504...
102528	0	-0.533...	-0.589...	-0.520...	-0.857...	-0.481...	-0.437...	-0.734...	-0.733...	-0.654...
103296	0	-0.470...	-0.459...	-0.651...	-0.656...	-0.768...	-0.422...	-0.679...	-0.746...	-0.684...
104064	0	-0.565...	-0.244...	-0.960...	-0.673...	-0.836...	-0.392...	-0.342...	-0.711...	-0.720...
104832	1	-0.690...	-0.557...	-0.838...	-0.947...	-0.859...	-0.618...	-0.576...	-0.535...	-0.620...
105600	1	-1.050...	-0.685...	-1.004...	-1.140...	-0.987...	-0.940...	-0.714...	-0.898...	-0.733...
106368	0	-0.612...	-0.510...	-1.306...	-0.787...	-0.675...	-0.618...	-1.069...	-0.617...	-0.793...
107136	1	-0.588...	-0.503...	-0.595...	-0.637...	-0.708...	-0.537...	-0.710...	-0.460...	-0.874...
107904	0	-0.530...	-0.444...	-0.570...	-0.820...	-0.554...	-0.507...	-0.390...	-0.564...	-0.394...
108672	1	-1.055...	-0.750...	-0.960...	-1.514...	-0.843...	-0.717...	-0.852...	-0.973...	-1.157...
109440	1	-0.498...	-0.509...	-0.706...	-0.585...	-0.601...	-0.686...	-0.661...	-0.475...	-0.780...

图 10-27　标签列与特征

至此，我们完成了训练数据的构造，预测数据的构造流程也与此相同，这里不再详细介绍。

10.2.2　模型训练

本方法生成的特征维度为 300，且都是数值类型，可以使用多种分类模型。整体的流程如图 10-28 所示。

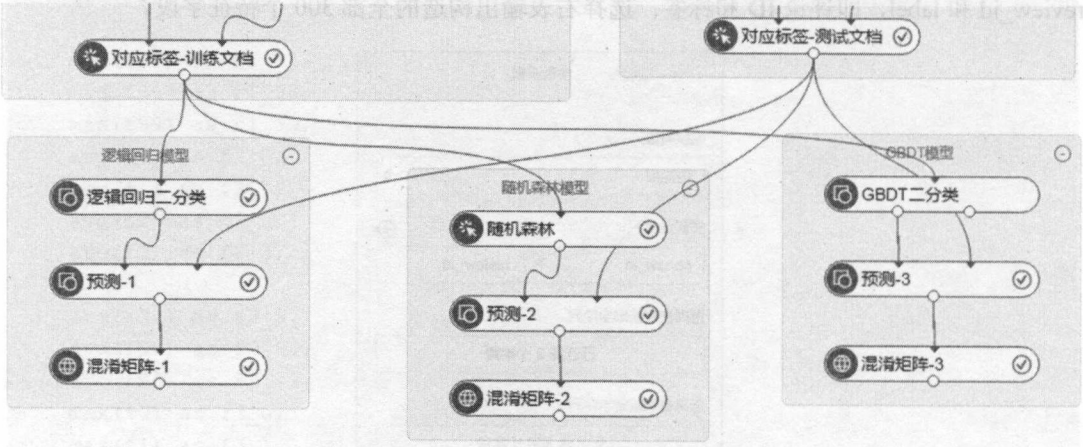


图 10-28　三种分类模型的训练、预测和评估

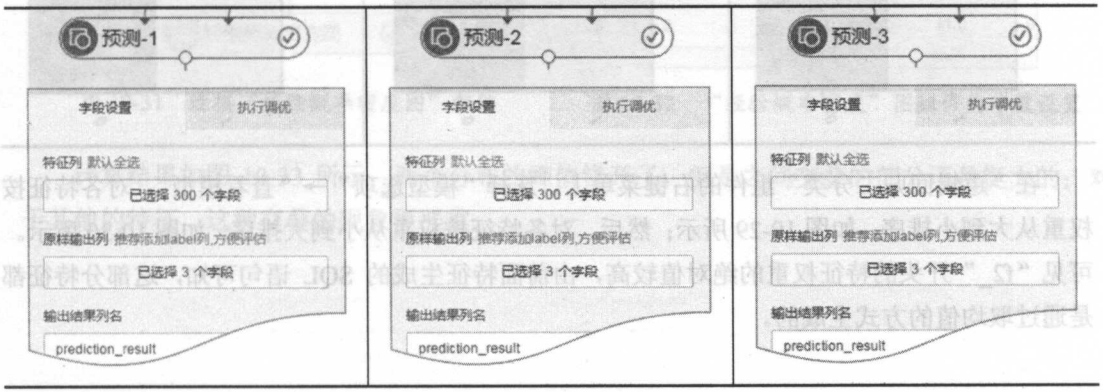
我们先关注各分类组件的设置，为了便于比较，列为表格，如表 10-2 所示，在“字段设置”标签栏，选择训练特征列为生成的 300 个特征字段；设置训练的目标列（标签列）为 label。

表 10-2 不同分类模型的字段参数设置



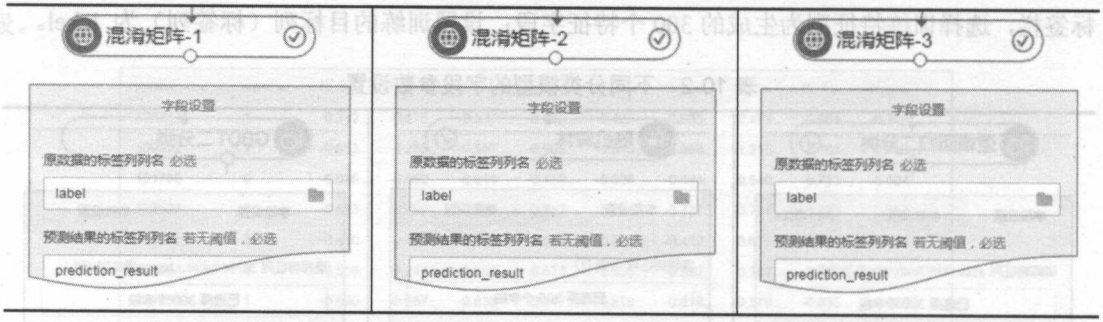
各个预测组件的设置相同，如表 10-3 所示，特征列选择生成的 300 个特征字段；原样输出 review_id、review 和 label 列，便于后面对预测结果进行分析和评估。

表 10-3 各模型预测组件的参数设置



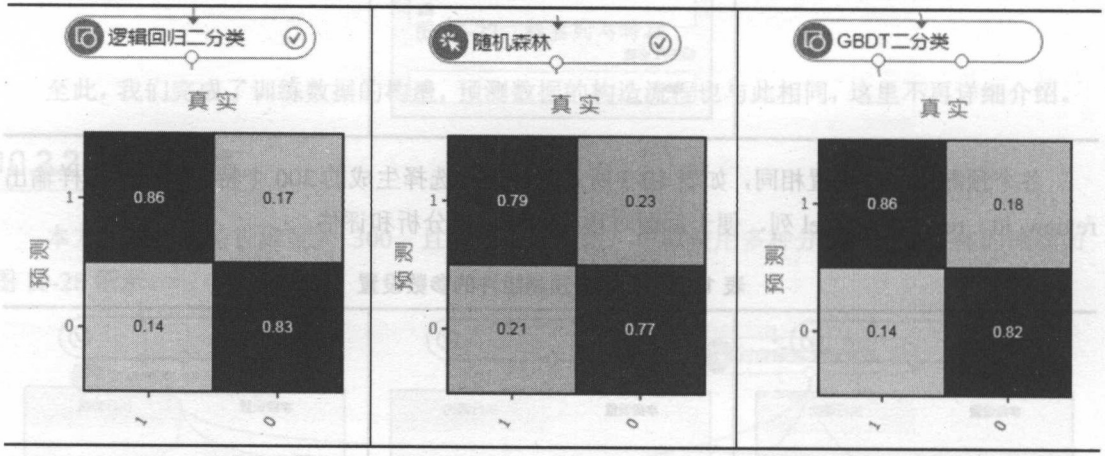
我们选择了比较简单的“混淆矩阵”组件进行评估，需要将原数据的标签列列名设置为 label，其他的使用默认设置即可，如表 10-4 所示。

表 10-4　各模型对应混淆矩阵组件的参数设置



这三种模型运行结果的比例矩阵如表 10-5 所示。作者也尝试设置随机森林和 GBDT 模型的训练参数，但效果上都没有超过逻辑回归模型，也许此训练数据的特点不能体现决策树类算法的优势。关于该数据的特点，稍后会从特征重要性及经验概率密度分布角度进行分析。

表 10-5　各模型评估结果的比例矩阵



在“逻辑回归二分类”组件的右键菜单上，选择“模型选项”→“查看模型”，对各特征按权重从大到小排序，如图 10-29 所示；然后，对各特征按权重从小到大排序，如图 10-30 所示。可见“f2_”开头的特征权重的绝对值较高，由前面特征生成的 SQL 语句可知，这部分特征都是通过取均值的方式生成的。



图 10-29 特征权重从大到小排列



图 10-30 特征权重数据从小到大排列

我们可以通过“经验概率密度图”组件，了解不同标签的分布的情况，如图 10-31 所示，将训练数据集连接到“经验概率密度图”组件，并如图 10-32 所示，设置标签列 label，指定需要计算的输入列，这里我们选择了权重绝对值最高的特征 f2_18 附近的 21 个特征。

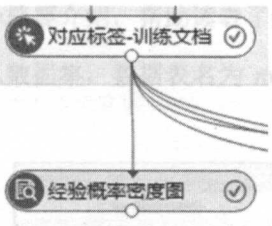


图 10-31 连接“经验概率密度图”组件

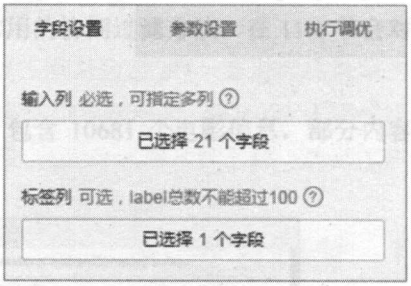


图 10-32 “经验概率密度”图组件的字段设置

计算结果如图 10-33 所示，两个分布的峰值错开了，但是之间交叠的部分还是挺大的；对于其他的特征，这种交叠的现象更严重。

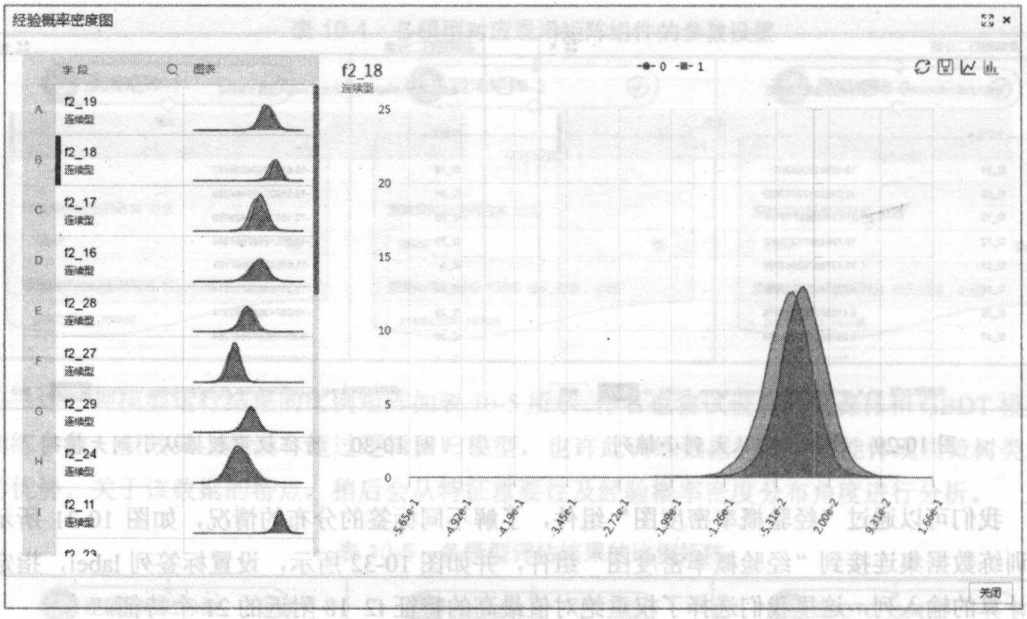


图 10-33 “经验概率密度图”组件的结果展示

第 11 章
影片推荐

本章讨论的主题是推荐，如何参考已知的数据，对每个用户所关注的商品或信息，给出个性化、精准的推荐。

本章会围绕一个电影推荐的具体场景展开，需要用到两个数据集：电影信息和用户对电影的评分记录。均来自 MovieLens 网站（http://movielens.org），下载地址为：http://grouplens.org/datasets/movielens/。使用的推荐算法为推荐领域比较常用的协同过滤方法，在 11.1 节会对该方法的原理有个简要介绍，帮助读者了解和使用该方法。

电影信息数据集，数据表名为 movielens_movies，包含 10681 个电影信息，部分内容如图 11-1 所示。

movieId	title	genres
1	Toy Story (1995)	Adventure\|Animation\|Children\|Comedy\|Fantasy
2	Jumanji (1995)	Adventure\|Children\|Fantasy
3	Grumpier Old Men (1995)	Comedy\|Romance
4	Waiting to Exhale (1995)	Comedy\|Drama\|Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action\|Crime\|Thriller
7	Sabrina (1995)	Comedy\|Romance
8	Tom and Huck (1995)	Adventure\|Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action\|Adventure\|Thriller
11	American President, The (1995)	Comedy\|Drama\|Romance
12	Dracula: Dead and Loving It (1995)	Comedy\|Horror
13	Balto (1995)	Animation\|Children
14	Nixon (1995)	Drama
15	Cutthroat Island (1995)	Action\|Adventure\|Romance
16	Casino (1995)	Crime\|Drama

图 11-1　数据表 movielens_movies 的内容

数据表共有三列，第一列名称为 movieid，即电影 ID，在下面的数据集中，通过电影 ID 关联到评论信息；第二列名称为 title，是电影名称，电影名称后的括号内为电影发行的年份；第三列名称为 genres，是电影类别（喜剧、动作片、爱情喜剧等）。

用户对电影的评分记录集，数据表名为 movielens_ratings，包含 10 000 054 条用户对电影的打分记录，内容如图 11-2 所示。

userid	movieid	rating
1	122	5
1	185	5
1	231	5
1	292	5
1	316	5
1	329	5
1	355	5
1	356	5
1	362	5
1	364	5

图 11-2　数据表 movielens_ratings 的内容

同前面的数据表一样，它也只有三列，第一列名称为 userid，用一个数字 ID 代表某个用户；第二列名称为 movieid，是电影 ID，在前面的数据集介绍中已经提过，可以使用此 ID 对应到具体的电影信息；第三列名称为 rating，是用户给该影片的打分，范围为 0.5, 1, 1.5, …, 4.5, 5。

11.1　协同过滤

协同过滤（Collaborative Filtering）是一种利用集体智慧的方法，从众多用户的历史行为中，收集目标用户相似的信息，从而发现用户潜在的兴趣偏好。

我们在日常生活中就在使用协同过滤的思想：你想看电影，可能会询问与自己品味相似的朋友，他们最近看了什么；自己喜欢的电影如果出了续集，也可能吸引你走进电影院。这就引出了两个常用的协同过滤方法：

1）基于用户（User-Based）的协同过滤。即找到与你相似的用户，他们喜欢的很有可能也是你喜欢的；可以从已经看过的影片历史记录中，寻找和你的记录相似的用户作为“相似用户”。

2）基于物品（Item-Based）的协同过滤。还是由历史记录出发，同时被更多人观看过的影片，相似度更高；根据你的观影记录，关联出相似度更高的影片进行推荐。

在实际处理中，用户与物品的关系用矩阵表示出来，如图 11-3 所示，用户有n个$u_1, u_2, \cdots, u_m$，物品有m个$i_1, i_2, \cdots, i_n$，如果用户u_a购买过物品i_j，则可在矩阵的第a行j列记录购买次数，没有购买行为的位置可以标记为 0，这样我们就得到了用户与物品关系的矩阵。

	i_1	i_2	$\cdots$	i_j	$\cdots$	i_n
u_1						
u_2						
$\vdots$						
u_a				1		
$\vdots$						
u_m						

图 11-3　用户与物品的关系矩阵

由用户与物品关系的矩阵，我们可以衡量出用户间的相似程度、物品间的相似程度。具体的方法是：对于每个用户，将其所在的矩阵的行看为其向量值，用户间的距离或相似程度可以由其对应的向量进行计算；对于每个物品，将其所在的矩阵的列看为其向量值，物品间的相似程度可以由其对应的向量进行计算。

向量间比较常用余弦相似度，公式如下：

$$\text{CosSim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|_2 \|\vec{b}\|_2} = \frac{\sum a_i b_i}{\sqrt{(\sum a_i^2)(\sum b_i^2)}}$$

如果我们忽略向量的具体数值，在意每个位置是否为 0，则可以使用集合的 Jaccard 相似度：

$$\text{JaccardSim}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

11.2　整体流程

我们针对 Movielens 数据集的影片评价记录，使用基于物品（Item-Based）的协同过滤方法，为用户进行影片推荐。整体推荐流程如图 11-4 所示。

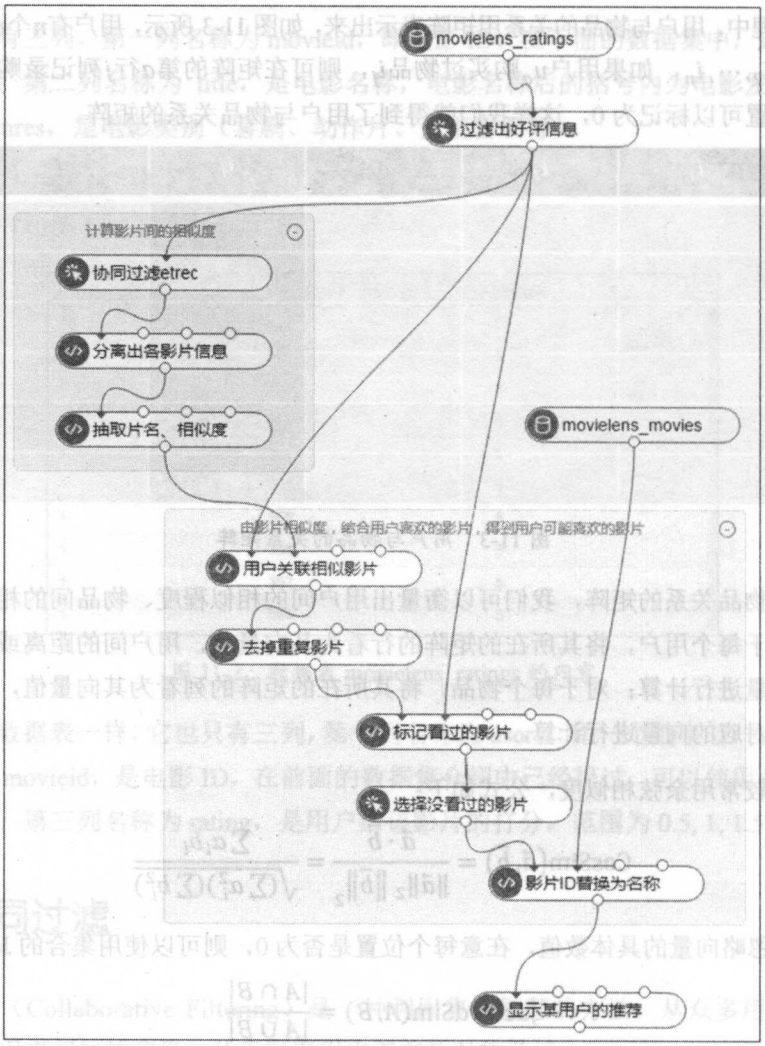


图 11-4　整体流程

我们需要用到两个数据集，主要是针对用户评分数据 movielens_ratings 计算出推荐方案，推荐的影片用 movieid 表示，为了方便查看推荐结果，使用数据集 movielens_movies，将电影 ID 转换为名称。

整个处理流程比较清晰，分为三个部分：

1）预处理，过滤出好评信息。

2）计算影片间的相似度。

3）计算用户可能喜欢的影片。

下面将对处理流程分别进行讲解分析，最后对比查看推荐结果，产生更具体的印象。

11.3 预处理，过滤出好评信息

在原始的评分数据中，有好评也有差评，由于我们的目标是给用户推荐可能喜欢的影片，这里采用比较简单有效的方法，就是直接过滤出好评信息，在此基础上进行后面的推荐。由于最高评分为 5 分，我们选择位于中等的 3 分作为标准，将高于 3 分的看作是好评信息。具体的操作方法是，选择“过滤与映射”组件，并设置其过滤条件为：rating>3，如图 11-5 所示。

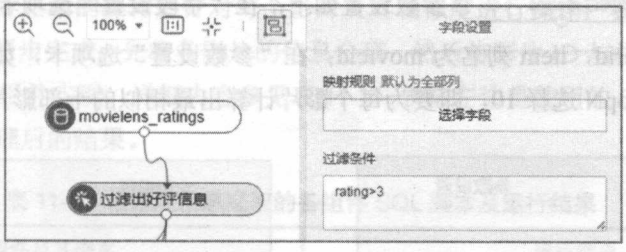


图 11-5 “过滤出好评信息”组件的参数设置

过滤后的数据如图 11-6 所示，将 rating 值大于 3 的那些好评数据过滤出来了。

userid	movieid	rating
3	6539	5
3	7153	4
3	7155	3.5
3	8529	4
3	8533	4.5
3	8783	5
3	27821	4.5
3	33750	3.5

图 11-6 “过滤出好评信息”组件的计算结果

11.4 计算影片间的相似度

该模块的流程如图 11-7 所示，第一个组件计算出相似度，后面两个组件进行格式变换，便于后面接 SQL 组件产生推荐结果。

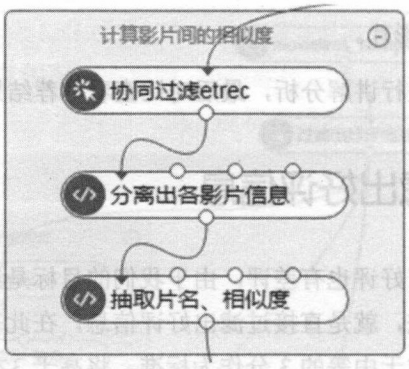
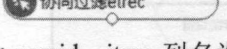


图 11-7　计算影片间的相似度

第一个组件  参数设置如下：在“字段设置”选项卡，如图 11-8 所示，选择 user 列名为 userid，item 列名为 movieid；在“参数设置”选项卡，如图 11-9 所示相似度类型选择 jaccard，TopN 选择 10，即要为每个影片计算出最相似的十部影片。

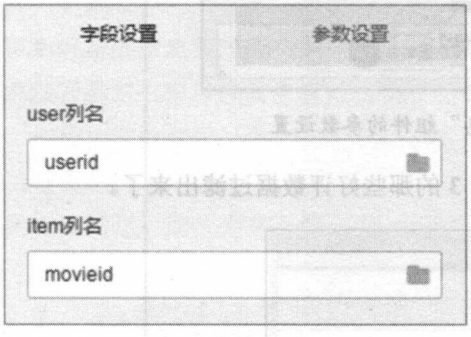


图 11-8　“协同过滤 etrec”组件的字段设置

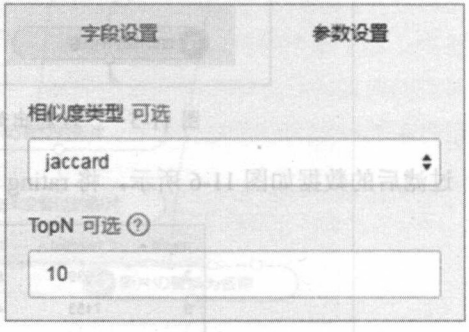


图 11-9　“协同过滤 etrec”组件的参数设置

计算结果如图 11-10 所示，第一列为影片 ID，第二列为前十部与影片相似的影片 ID 及相似度，各影片间用空格作为间隔，影片与相似度间使用冒号“:”作为间隔。

itemid	similarity
100	14:0.08830121022 79:0.07784773898 81:0.07128446537 707:0.07049965777 640:0.06865177833 628:0.06362410072 ...
1017	1013:0.1476306197 1012:0.1412708205 1019:0.130989011 1031:0.1197452229 1014:0.1160471442 1025:0.112991538...
1105	710:0.0976744186 842:0.09604519774 626:0.07303370787 891:0.07012195122 810:0.06179775281 725:0.0578512396...
1156	49355:0.1428571429 26431:0.1428571429 6113:0.125 7953:0.1111111111 7523:0.1111111111 60411:0.1111111111 54196...
1444	1058:0.08 1134:0.06382978723 843:0.05357142857 1439:0.04838709677 1532:0.04761904762 1443:0.04651162791 11...
1453	1474:0.08035714286 1043:0.06534090909 840:0.06222222222 1427:0.05803571429 881:0.05263157895 1457:0.05169...
1509	877:0.06578947368 4252:0.04166666667 3046:0.04 1897:0.03682719547 7615:0.03636363636 49:0.03636363636 299...
1578	1102:0.08333333333 1525:0.02564102564 8982:0.02173913043 1497:0.01449275362 1427:0.01265822785 692:0.0101...
1659	1708:0.05882352941 1740:0.05555555556 1738:0.05555555556 1510:0.05 1720:0.04761904762 2244:0.04761904762 ...
1879	2837:0.0775862069 2626:0.05479452055 1502:0.05303030303 2029:0.05263157895 2437:0.05244755245 4261:0.0465...

图 11-10　“协同过滤 etrec”组件的计算结果

由于后面生成推荐结果的操作会针对每个影片及其相似度进行操作，我们需要把这些内容都抽取出来，具体分两步完成：先将各影片的信息分离；然后将影片 ID 与相似度分离。每一步都使用“SQL 脚本”组件实现，详细内容参见表 11-1，表格左边是使用的 SQL 组件及其脚本，右边是经过此组件处理后的结果。

表 11-1　相似度信息提取的各组件 SQL 脚本及运行结果

SQL 组件及其脚本	运行结果
分离出各影片信息 **SQL 脚本：** select trans_array(1, " ", itemid, similarity) as (movieid, similarity) from ${t1}	movieid / similarity 1 / 260:0.3220215463 1 / 1210:0.2882281763 1 / 1270:0.2793811703 1 / 1198:0.2791872902 1 / 1196:0.276462324 1 / 356:0.2689287557 1 / 3114:0.2670725376 1 / 318:0.265466663 1 / 588:0.2652499321 1 / 2571:0.2562545867
抽取片名、相似度 **SQL 脚本：** select movieid, split_part(similarity, ":", 1) as similar_movieid, split_part(similarity, ":", 2) as similarity from ${t1}	movieid / similar_movieid / similarity 1 / 260 / 0.3220215463 1 / 1210 / 0.2882281763 1 / 1270 / 0.2793811703 1 / 1198 / 0.2791872902 1 / 1196 / 0.276462324 1 / 356 / 0.2689287557 1 / 3114 / 0.2670725376 1 / 318 / 0.265466663 1 / 588 / 0.2652499321 1 / 2571 / 0.2562545867

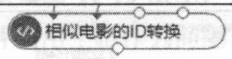
至此，我们已经完成了该部分三个组件的介绍。此时的计算结果中影片是用 ID 表示的，我们看不到任何实际意义，下面我们对该部分的输出结果进行一系列的 ID 转化，看看具体是哪些影片。如图 11-11 所示，在原有的流程中开出一个分支，在图中用加粗的连线显示，使用两个 SQL 组件进行 ID 转换，这里需要用到数据集 movielens_movies。



图 11-11　电影 ID 与电影名的转换

所涉及的两个处理步骤都使用“SQL 脚本”组件实现，各组件的 SQL 脚本及其运行结果如表 11-2 所示。

表 11-2　电影 ID 与名称转换的 SQL 组件脚本及运行结果

SQL 组件及其脚本	运行结果
 **SQL 脚本：** SELECT a.movieid, b.title AS similar_moviename, a.similarity FROM ${t1} a JOIN ${t2} b ON a.similar_movieid =b.movieid	movieid ▲ / similar_moviename ▲ / similarity ▲ 852 / Grumpier Old Men (1995) / 0.09273570325 79 / Grumpier Old Men (1995) / 0.0949687407 788 / Grumpier Old Men (1995) / 0.1295499022 637 / Grumpier Old Men (1995) / 0.08314534322 762 / Grumpier Old Men (1995) / 0.0677378702 719 / Grumpier Old Men (1995) / 0.09451575263 135 / Grumpier Old Men (1995) / 0.08982584785 3450 / Grumpier Old Men (1995) / 0.0949171581 5 / Grumpier Old Men (1995) / 0.1451811684 555 / Heat (1995) / 0.1832753795

续表

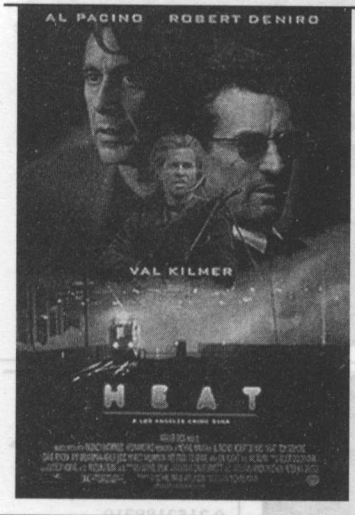
目标电影的ID转换

SQL 脚本：

```
SELECT b.title AS moviename, a.similar_moviename, a.similarity FROM ${t1} a JOIN ${t2} b ON a.movieid =b.movieid
```

moviename	similar_moviename	similarity
Heat (1995)	Reservoir Dogs (1992)	0.1944653614
Heat (1995)	12 Monkeys (Twelve Monkeys) (1995)	0.205739346
Heat (1995)	Seven (a.k.a. Se7en) (1995)	0.2164417529
Heat (1995)	Usual Suspects, The (1995)	0.2047025707
Heat (1995)	Die Hard (1988)	0.1956612932
Heat (1995)	Rock, The (1996)	0.2152188719
Heat (1995)	Terminator 2: Judgment Day (1991)	0.1840565578
Heat (1995)	Fargo (1996)	0.1852014395
Heat (1995)	Casino (1995)	0.2119873291
Heat (1995)	Léon: The Professional (Léon) (Professional, T...	0.2050528004

我们以影片 Heat（1995）为例，查看其详细信息如下。



影片名称：Heat（1995）

内容简介：

该片讲述了两个劫匪麦考利和克里斯在一场惊天大劫案中与洛杉矶警察在街头展开决战的故事。

类　　型：

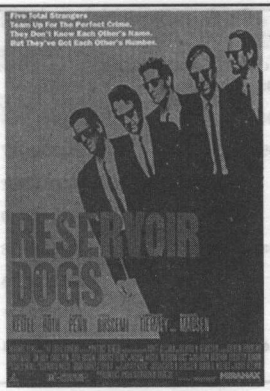
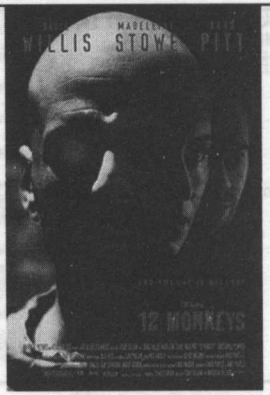
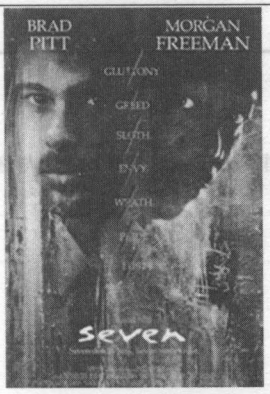
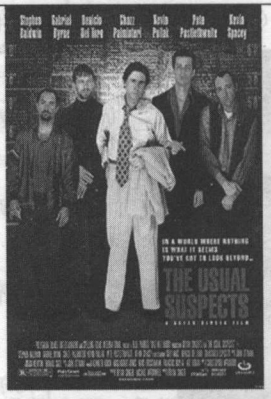
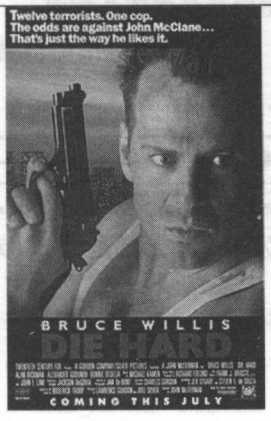
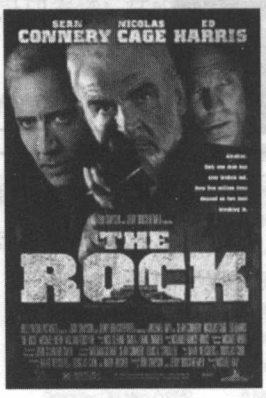
动作、犯罪、剧情、惊悚

主　　演：

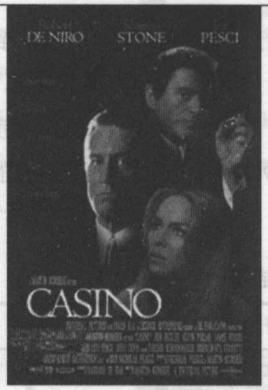
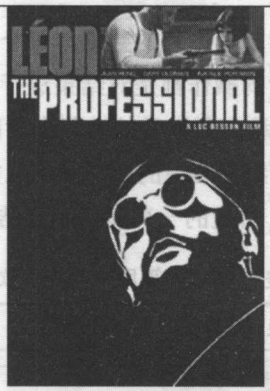
阿尔·帕西诺，罗伯特·德尼罗，方·基默，强·沃特

将与之相似的前十部影片的名称、相似度和海报图片整理到表 11-3。有些影片我们看过，只是不熟悉其英文名，有的影片我们没有看过，但查一下影片简介，发现基本上都是属于“动作、犯罪、剧情、惊悚”类型的。

表 11-3　计算出的 10 部相似影片

	影片名称： Reservoir Dogs (1992) 相似度： 0.1944653614		影片名称： 12 Monkeys (Twelve Monkeys) (1995) 相似度： 0.205739346
	影片名称： Seven (Se7en) (1995) 相似度： 0.2164417529		影片名称： Usual Suspects, The (1995) 相似度： 0.2047025707
	影片名称： Die Hard (1988) 相似度： 0.1956612932		影片名称： Rock, The (1996) 相似度： 0.2152188719

续表

	影片名称： Terminator 2: Judgment Day (1991) 相似度： 0.1840565578		影片名称： Fargo (1996) 相似度： 0.1852014395
	影片名称： Casino (1995) 相似度： 0.2119873291		影片名称： Léon: The Professional (Léon) (Professional, The) (1994) 相似度： 0.2050528004

11.5　计算用户可能喜欢的影片

在前面计算出的相似度基础上，本小节要为每个用户产生最终的推荐结果，这部分的流程如图 11-12 所示。

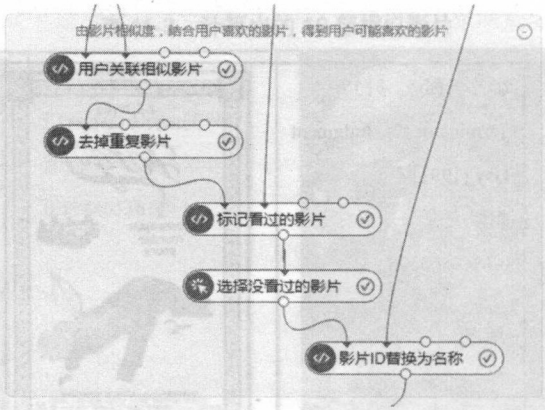
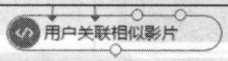


图 11-12　后处理，得到用户可能喜欢的影片

第一个组件将用户观看影片的数据与每个影片的相似影片数据进行连接，实现了从用户到相似影片的对应；随后的操作都是对结果进行优化，去掉重复的内容，将看过的影片过滤掉，并将影片 ID 替换为名称。所涉及的过程中有四步是使用“SQL 脚本”组件实现的，另外一步是通过“过滤与映射”组件实现的，详细信息如表 11-4 所示，左边一列显示组件名称及相应的 SQL 语句或参数配置，右边的一列显示运行结果的截图。

表 11-4　计算用户可能喜欢影片的各组件设置及运行结果

组件及其配置	运行结果
 **SQL 脚本：** SELECT a.userid, a.movieid, b.similar_movieid, b.similarity FROM ${t1} a JOIN ${t2} b ON a.movieid=b.movieid	userid ▲ \| movieid ▲ \| similar_movieid ▲ \| similarity ▲ 57655 \| 3 \| 95 \| 0.0991062879 57655 \| 3 \| 736 \| 0.1091438817 57655 \| 3 \| 7 \| 0.1098513011 57655 \| 3 \| 494 \| 0.1124420402 57655 \| 3 \| 62 \| 0.1126136364 57655 \| 3 \| 786 \| 0.1162833487 57655 \| 3 \| 802 \| 0.1170005318 57655 \| 3 \| 104 \| 0.1225759768 57655 \| 3 \| 788 \| 0.1295499022 57655 \| 3 \| 5 \| 0.1451811684

续表

去掉重复影片

SQL 脚本：

```
SELECT userid, similar_movieid, max(similarity) AS similarity FROM ${t1} GROUP BY userid, similar_movieid
```

userid	similar_movieid	similarity
1	1	0.2652499321
1	2	0.1605582524
1	4	0.06738868833
1	10	0.2041140437
1	11	0.2453654983
1	19	0.1772116065
1	20	0.05876068376
1	23	0.06443719413
1	47	0.159753677
1	48	0.09463198672

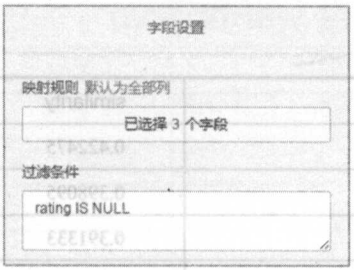
标记看过的影片

SQL 脚本：

```
SELECT a.*, b.rating FROM ${t1} a LEFT OUTER JOIN ${t2} b ON a.userid=b.userid AND a.similar_movieid=b.movieid
```

userid	similar_movieid	similarity	rating
1	2	0.1605582524	-
1	11	0.2453654983	-
1	20	0.058760683...	-
1	23	0.064437194...	-
1	47	0.159753677	-
1	104	0.1646038295	-
1	110	0.3866064929	-
1	158	0.1024078909	-
1	161	0.2512961508	-
1	173	0.0951863916	-

选择没看过的影片



similarity	userid	similar_movieid
0.09463198672	1	48
0.05821917808	1	93
0.3677357499	1	150
0.273825121	1	165
0.1164359247	1	204
0.1614322561	1	225
0.05702917772	1	234
0.07135969142	1	267
0.1217736598	1	315
0.3980946349	1	318

续表



SQL 脚本：

```
SELECT a.userid, b.title AS recommend_moviename,
a.similarity FROM ${t1} a JOIN ${t2} b ON
a.similar_movieid=b.movieid
```

userid	recommend_moviename	similarity
71324	Grumpier Old Men (1995)	0.09451575263
21325	Grumpier Old Men (1995)	0.1295499022
34789	Grumpier Old Men (1995)	0.09451575263
30263	Grumpier Old Men (1995)	0.0949687407
59483	Grumpier Old Men (1995)	0.1295499022
8849	Grumpier Old Men (1995)	0.09273570325
42958	Grumpier Old Men (1995)	0.09273570325
66883	Grumpier Old Men (1995)	0.1451811684
49552	Grumpier Old Men (1995)	0.09273570325
12868	Grumpier Old Men (1995)	0.0949687407

从此结果表中，我们可以查询到每一个用户的推荐影片，并可根据相似度确定推荐的顺序。下一节，会以一个用户为例，查看推荐的效果。

11.6 查看推荐效果

工作流中的SQL组件 

即可完成推荐结果的查看，具体的SQL脚本如下，只要将 userid 的值修改为我们需要查询的 id，这里我们选择 userid=1 的用户。

```
select * from ${t1} where userid='1' order by similarity desc limit 100
```

我们截取显示相似度较大的结果如表 11-5 所示。

表 11-5 推荐影片列表

userid	recommend_moviename	similarity
1	Beauty and the Beast (1991)	0.422475
1	Shawshank Redemption, The (1994)	0.398095
1	Silence of the Lambs, The (1991)	0.391333
1	Fugitive, The (1993)	0.389718
1	Braveheart (1995)	0.386606
1	Terminator, The (1984)	0.38044
1	Apollo 13 (1995)	0.367736
1	Pulp Fiction (1994)	0.363414
1	Schindler's List (1993)	0.352646

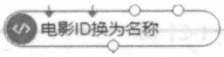
续表

userid	recommend_moviename	similarity
1	Matrix, The (1999)	0.337066
1	Pretty Woman (1990)	0.334843
1	Star Wars: Episode V - The Empire Strikes Back (1980)	0.330774
1	Dances with Wolves (1990)	0.328695
1	Ace Ventura: Pet Detective (1994)	0.32341
1	Ghost (1990)	0.315139
1	True Lies (1994)	0.297746
1	Mrs. Doubtfire (1993)	0.295463
1	Pinocchio (1940)	0.293252
1	While You Were Sleeping (1995)	0.275286
1	Die Hard: With a Vengeance (1995)	0.273825
1	Toy Story (1995)	0.26525

当然了，我们还需要了解该用户喜欢什么样的影片，才知道推荐的内容是否合适。我们可以通过搭建一个小流程来实现，如图 11-13 所示，使用两个 SQL 组件，分别完成电影 ID 转换为名称和将 userid=1 的用户记录过滤出来。

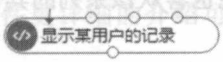


图 11-13 显示某用户的记录

其中，SQL 组件 

的脚本为

```
SELECT a.userid, b.title AS moviename, a.rating FROM ${t1} a JOIN ${t2} b ON a.movieid=b.movieid
```

另一个 SQL 组件 

的脚本为

```
select * from ${t1} where userid='1' order by moviename desc limit 100
```

运行结果如表 11-6 所示。

表 11-6　用户（userid=1）喜欢的影片列表

userid	moviename	rating
1	Terminator 2: Judgment Day (1991)	5
1	Stargate (1994)	5
1	Star Trek: Generations (1994)	5
1	Speed (1994)	5
1	Snow White and the Seven Dwarfs (1937)	5
1	Sleepless in Seattle (1993)	5
1	Robin Hood: Men in Tights (1993)	5
1	Outbreak (1995)	5
1	Net, The (1995)	5
1	Naked Gun 33 1/3: The Final Insult (1994)	5
1	Lion King, The (1994)	5
1	Jurassic Park (1993)	5
1	Jungle Book, The (1994)	5
1	Hot Shots! Part Deux (1993)	5
1	Home Alone (1990)	5
1	Forrest Gump (1994)	5
1	Flintstones, The (1994)	5
1	Dumb & Dumber (1994)	5
1	Boomerang (1992)	5
1	Beverly Hills Cop III (1994)	5
1	Aristocats, The (1970)	5
1	Aladdin (1992)	5

针对这两张结果表，我们可以发现用户喜欢看 Snow White and the Seven Dwarfs (1937)、Lion King, The (1994)、Aladdin (1992)，则对同一类型的动画片 Beauty and the Beast (1991)也应该比较喜欢。喜欢 Terminator 2: Judgment Day (1991)，很可能也喜欢该系列的其他影片 Terminator, The (1984)。总体来说，推荐效果还是不错的。

第12章 支持深度学习框架

阿里云机器学习算法平台除了提供自己研发的组件，也支持业界一些出名的深度学习框架，同时后端提供了功能强大的 GPU（型号 M40）计算集群，用户可以使用这些框架及硬件资源来运行深度学习算法。目前支持的框架包括 TensorFlow r0.12，MXNet 0.9.2，Caffe RC3。

本章重点介绍 Google 开源的机器学习框架 TensorFlow。TensorFlow 主要针对机器学习和深度神经网络方面的算法，通过有向图来描述算法，并能通过 CPU 或 GPU 对描述的算法进行高效地计算。有向图的节点表示操作（譬如：数据输入、矩阵乘、梯度下降优化等），每个节点计算产生的结果为多维数据数组（即张量，Tensor），通过有向边传给其他的节点。我们可以想象到运行的时候，张量（Tensor）数据在有向图中流动（Flow）。

12.1 TensorFlow 组件简介

调用 TensorFlow 任务，需要用到两个组件，其连接方式如图 12-1 所示。

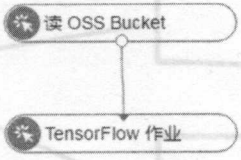


图 12-1 TensorFlow 组件连接

其中，“读 OSS Bucket”组件用来设置 TensorFlow 任务在读、写用户 OSS Bucket 时使用的授权信息及域名；组件“TensorFlow 作业”负责提交脚本，设置参数，运行任务，查看日志等工作。关于这两个组件更详细的介绍可以参考帮助文档：https://help.aliyun.com/document_detail/49571.html。

我们将要处理的 iris 数据集，数据量较少，可以随任务脚本直接上传提交，我们可以跳过申请 OSS 权限的步骤，直接进入主题，运行 TensorFlow 任务。

数据集 iris 的介绍，详见本书第 6 章，这里使用的数据集是从 TensorFlow 网站的演示例子中下载的，具体方法是从 https://github.com/tensorflow/tensorflow 下载整个项目的源码，解压后，通过路径 tensorflow/examples/tutorials/monitors/ 便可找到两个数据文件：iris_tese.csv 和 iris_training.csv。

我们准备运行的脚本文件：iris.py，先实现最基本的功能，从 csv 文件中读取数据，并把 train_set 打印出来。详细代码如下：

```
import numpy as np
import tensorflow as tf
train_set = tf.contrib.learn.datasets.base.load_csv_with_header(
    filename="iris_training.csv",                    target_dtype=np.int,
features_dtype=np.float)
test_set = tf.contrib.learn.datasets.base.load_csv_with_header(
    filename="iris_test.csv", target_dtype=np.int, features_dtype=np.float)
print(train_set)
```

现在，我们手头已经有三个文件：iris.py，iris_training.csv 和 iris_test.csv。在上传提交给组件执行前，还需要将它们打包压缩为一个文件，iris.tar.gz，如图 12-2 所示。

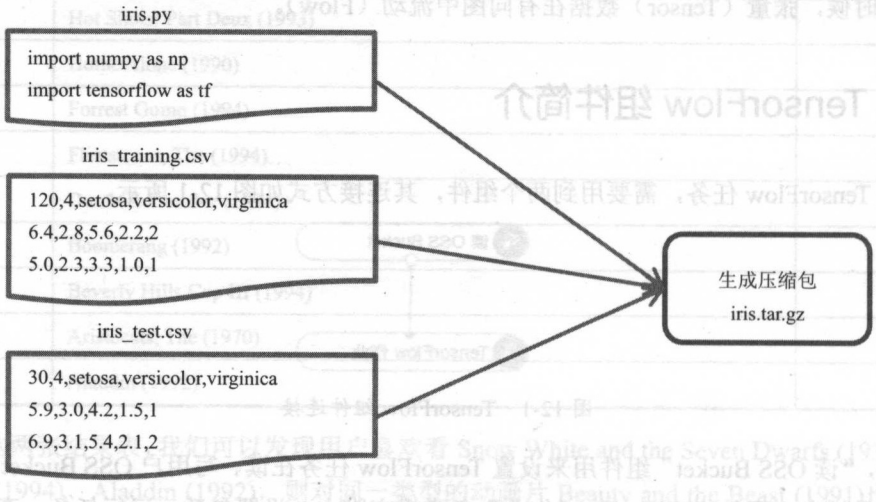


图 12-2 脚本、数据文件压缩示意图

在“TensorFlow 作业”组件的“特征选择”选项卡中，需要设置前两项参数，如图 12-3 所示，在“Python 代码文件”输入框中，单击云状的图标，即可选择 iris.tar.gz 文件的路径，完成上传；在“Python 主文件”输入框中，设置 Python 脚本的文件名为 iris.py。

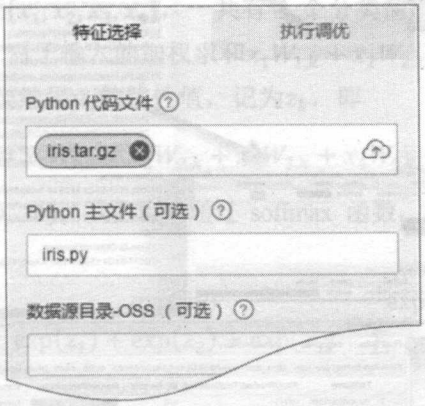


图 12-3　“TensorFlow”作业组件的参数配置

完成上面的设置，便可以选择运行“TensorFlow 作业”组件。在运行的过程中或者结束后，我们可以查看计算任务的打印输出及 TensorFlow 的日志输出，具体的步骤如图 12-4 所示，选择组件的“查看日志”菜单项；在弹出的“查看日志”窗体中，单击 logview 的链接；出现该计算任务的整体信息，然后再单击该计算任务；出现该计算任务中具体计算节点的信息，注意该页面上圈中的两个区域：StdOut 和 StdErr；Python 代码中打印输出的内容会显示在 StdOut 上，而 TensorFlow 的日志会输出到 StdErr。

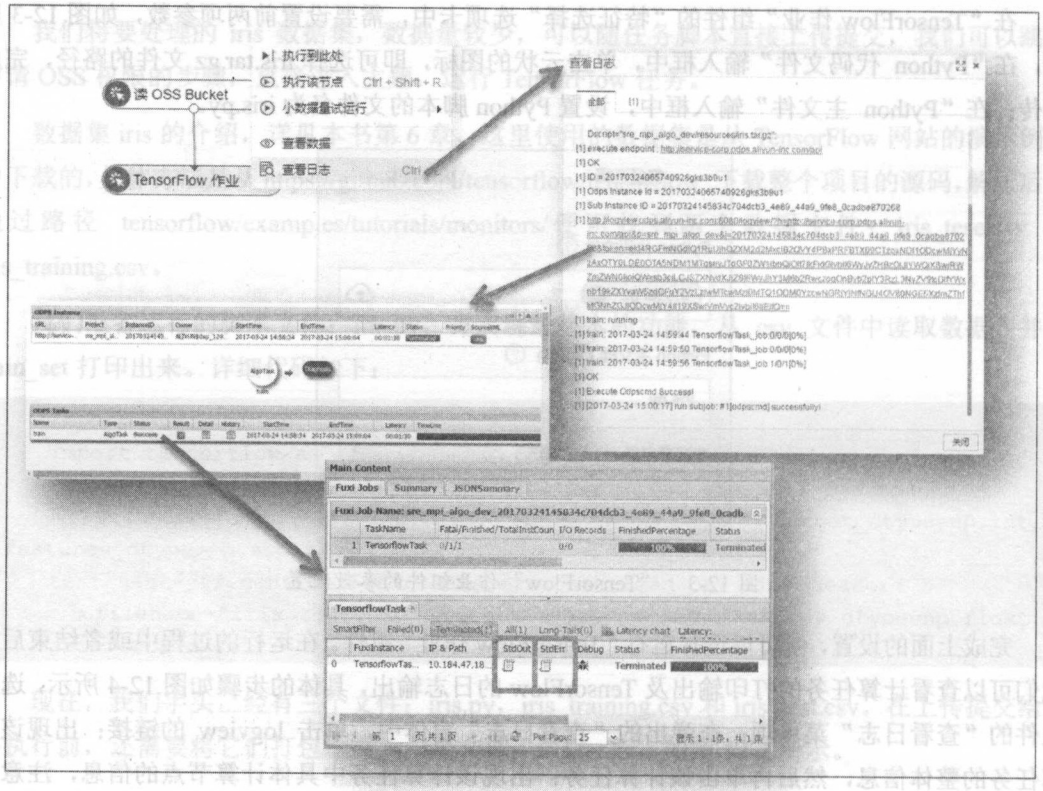


图 12-4　TensorFlow 组件运行结果查看

打印输出的内容如下，由于内容较多，两处用了省略号代替中间的数据。可以看到 train_set 下面包括两个数组：data 和 target。data 是个二维数组，每行是一个记录，有 4 个属性；target 是一维数组，0、1、2 分别为分类的索引号。

```
Dataset(data=array([[ 6.4,  2.8,  5.6,  2.2],
       [ 5. ,  2.3,  3.3,  1. ],
       [ 4.9,  2.5,  4.5,  1.7],
       ......
       [ 4.8,  3. ,  1.4,  0.1],
       [ 5.5,  2.4,  3.7,  1. ]]),
       target=array([2, 1, 2, ....., 0, 0, 1])
)
```

成功读取数据后，我们就可以开始设计训练流程了。

12.2　Softmax 模型

要求解的问题本身是个多分类问题，我们先使用 softmax 模型来处理，模型的原理如图 12-5 所示。数据有 4 个特征$\{x_1, x_2, x_3, x_4\}$，一共有 3 个分类情况$\{y_1, y_2, y_3\}$，每个特征x_i对分类结果y_k的贡献权重为$W_{i,k}$，对于输入的加权求和$x_1W_{1,k} + x_2W_{2,k} + x_3W_{3,k} + x_4W_{4,k}$，再分别加上一个偏置项$b_k$，是关于分类结果$y_k$的线性值，记为$z_k$，即

$$z_k = x_1W_{1,k} + x_2W_{2,k} + x_3W_{3,k} + x_4W_{4,k} + b_k$$

此时z_k的取值范围为$(-\infty, \infty)$，我们可以通过 softmax 函数，将$\{z_1, z_2, z_3\}$变换为$\{y_1, y_2, y_3\}$，具体方法如下：

$$y_k = \frac{\exp(z_k)}{\exp(z_1) + \exp(z_2) + \exp(z_3)} = \frac{\exp(z_k)}{\sum_j \exp(z_j)}$$

显然，

$$0 \le y_k \le 1 \text{ 且 } y_1 + y_2 + y_3 = 1$$

这样得到的结果$\{y_1, y_2, y_3\}$可以看作是属于各个分类的概率值。

其实，softmax 函数没有改变各数值间的大小顺序，如果z_k为$\{z_1, z_2, z_3\}$中的最大值，则y_k也为$\{y_1, y_2, y_3\}$中最大值，使用 softmax 函数除了使结果更规范外，还可以使模型参数训练时更快收敛，从而加快模型训练速度。

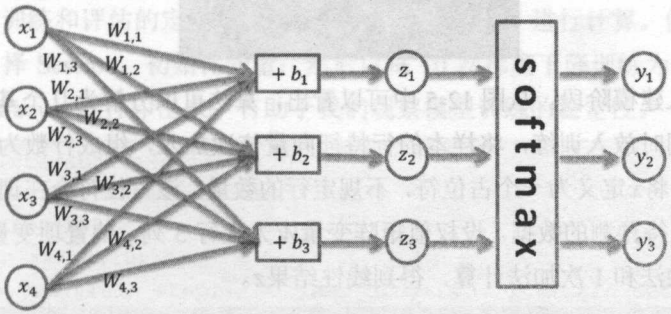


图 12-5　softmax 模型

在建模前，我们还要对数据进行格式变换，将分类索引值转化为 one-hot 格式，即用一个程度为分类数的向量表示分类结果，其中，分类索引对应的位置的值为 1，其余为 0，具体实现中，可以使用 TensorFlow 的 one-hot 函数，该函数的链接为：

https://www.tensorflow.org/api_docs/python/tf/one_hot。

注意：调用 tf.one-hot(...)函数返回的结果为 Tensor 对象，直接 print 只能打印输出其 shape 的信息，还需要使用 eval()函数得到其数组的值。

下面的代码实现了整个变换。

```
for i in range(5):
    print(train_set.target[i])
with tf.Session():
    train_target_one_hot = tf.one_hot(train_set.target, 3, on_value=1,
off_value=0).eval()
    test_target_one_hot = tf.one_hot(test_set.target, 3, on_value=1,
off_value=0).eval()
for i in range(5):
    print(train_target_one_hot[i])
```

运行的结果如下，前 5 个是行为索引值，后面的是 5 个行为相应的 one-hot 结果。

```
2
1
2
0
0
[0 0 1]
[0 1 0]
[0 0 1]
[1 0 0]
[1 0 0]
```

下面真正进入建模阶段。从图 12-5 中可以看出，算法可以分解为几个基本运算，并考虑到将训练样本数据同时放入训练，将样本的行特征向量依次加入，组成行数为样本个数，列数为特征维度的矩阵。将x定义为一个占位符，不规定行的数量，该占位符会在训练的时候放上训练数据，预测时放入待预测的数据。设权值矩阵变量W为 4 行 3 列，偏置项变量b为长度为 3 的行向量，通过 1 次乘法和 1 次加法计算，得到线性结果z。

```
feature_num = 4
target_num = 3
x = tf.placeholder(tf.float32, [None, feature_num])
W = tf.Variable(tf.zeros([feature_num, target_num]))
b = tf.Variable(tf.zeros([target_num]))
z = tf.matmul(x, W) + b
```

中间的线性结果z到目标结果y，还要经过一个 softmax 计算。在 TensorFlow 实际的应用中，不用将这个计算显性表示出来，可以将其和损失函数定义在一起。y为 one-hot 格式的占位符，行数为样本个数，列数为分类个数的矩阵。使用交叉熵（cross entropy）作为损失函数，使用最优化算法逐步求解参数W和b，使损失函数值最小。在实际应用中，可以选择 TensorFlow 自带的交叉熵函数 tf.nn.softmax_cross_entropy_with_logits，需要注意的是，函数中的 label 参数为 one-hot 格式的真实的分类值，参数 logits 是经过W和b线性计算后的结果（即z）；在模型逐步训练的过程中，采用梯度下降算法 tf.train.GradientDescentOptimizer。

```
    y = tf.placeholder(tf.float32, [None, target_num])
    cross_entropy                                                    =
tf.reduce_sum(tf.nn.softmax_cross_entropy_with_logits(logits=z, labels=y))
    train_step                                                       =
tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)
```

然后，再定义评估函数，分为两步进行，先计算出每条记录的预测值与真实值是否相同；然后，将布尔值 True 转化为 1，False 转化为 0，计算出的平均值即为精确度（Accuracy）。

```
    correct_prediction = tf.equal(tf.argmax(z, 1), tf.argmax(y, 1))
    accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
```

其中函数 tf.argmax(…)可以给出指定 Tensor 对象在某一维度上数据值最大的项所在的索引号；函数 tf.equal(…)用来判断两个组数对应项的值是否一致，返回布尔型数组；函数 tf.reduce_mean(…)用来计算出数组内各元素的平均值。

有了上面关于训练和评估的定义，我们可以将其组织起来，进行计算。使用 TensorFlow 最常用的步骤，先选择 Session，初始化变量；然后以每 10 次梯度下降训练为一个单位，计算当时的精确度（Accuracy）并打印出来，有助于我们观察模型训练的稳定性。

```
    sess = tf.InteractiveSession()
    tf.global_variables_initializer().run()
    for k in range(100):
        for _ in range(10):
            sess.run(train_step,      feed_dict={x:      train_set.data,      y:
train_target_one_hot})
        cur_accuracy  =  sess.run(accuracy,  feed_dict={x:  test_set.data,  y:
test_target_one_hot})
        print("Step:" + str(k*10+10) + "\tAccuracy:" + str(cur_accuracy))
```

运行结果如下，后面输出的重复内容被省略。可以看到，迭代 200 次以后，精确度（Accuracy）已经稳定在 0.966667。

```
Step:10Accuracy:0.366667
Step:20Accuracy:0.866667
Step:30Accuracy:0.533333
Step:40Accuracy:0.733333
Step:50Accuracy:0.533333
Step:60Accuracy:0.733333
Step:70Accuracy:0.533333
Step:80Accuracy:0.533333
Step:90Accuracy:0.966667
Step:100Accuracy:0.733333
Step:110Accuracy:0.7
Step:120Accuracy:0.933333
Step:130Accuracy:0.733333
Step:140Accuracy:0.966667
Step:150Accuracy:0.633333
Step:160Accuracy:0.933333
Step:170Accuracy:0.466667
Step:180Accuracy:0.966667
Step:190Accuracy:0.966667
Step:200Accuracy:0.966667
Step:210Accuracy:0.966667
Step:220Accuracy:0.966667
.....
```

最后，我们构造两个特征，预测一下结果。需要用到 TensorFlow 的 feed 机制，简单地说，就是用新定义的数组给占位符x赋值，从而计算出线性结果z，前面的讨论中已经提到了，softmax 函数不会改变各数值间的大小关系，所以可由z值直接预测分类结果。具体代码如下：

```
    new_samples = np.array( [[6.4, 3.2, 4.5, 1.5], [5.8, 3.1, 5.0, 1.7]],
dtype=float)
    pred = sess.run(tf.argmax(z, 1), feed_dict={x: new_samples})
    print("Predictions: {}".format(str(pred)))
```

打印输出的结果如下，预测第一条记录的类别索引号为 1，第二条的类别索引号为 2。

```
Predictions: [1, 2]
```

12.3 深度神经网络

使用 TensorFlow 上的函数 tf.contrib.learn.DNNClassifier，可以方便地构建深度神经网络。参数 feature_columns 指定输入层节点的个数，每个数值特征作为一个输入节点，有 4 个；共使用三个隐层，每层分别有 10，20 和 10 个节点，各层间都是全连接方式；输出层节点的个数为

分类的个数，有 3 个。具体代码如下：

```
    classifier = tf.contrib.learn.DNNClassifier(
        feature_columns=[tf.contrib.layers.real_valued_column("",
dimension=4)],
        hidden_units=[10, 20, 10],
        n_classes=3)
```

模型训练也非常简单，只需要指定训练集的特征和分类，并指定训练迭代次数为 2000 次。

注意：这里的分类值为原始输入的索引值的形式，不需要 one-hot 的格式。

```
    classifier.fit(x=train_set.data, y=train_set.target, steps=2000)
```

在使用 DNN 模型时，如果需要输入训练过程中 loss 的变化，可以使用如下语句，输出 tf.logging.INFO 级别的日志。

```
    tf.logging.set_verbosity(tf.logging.INFO)
```

运行 TensorFlow 组件，我们可以在 StdErr 中看到 INFO 的信息，具体内容如下：

```
INFO:tensorflow:Saving checkpoints for 1 into
/tmp/tmpk_kAvv/model.ckpt.
INFO:tensorflow:loss = 2.16771, step = 1
INFO:tensorflow:global_step/sec: 572.941
INFO:tensorflow:loss = 0.226035, step = 101
INFO:tensorflow:global_step/sec: 608.239
INFO:tensorflow:loss = 0.107678, step = 201
INFO:tensorflow:global_step/sec: 608.125
INFO:tensorflow:loss = 0.0809396, step = 301
INFO:tensorflow:global_step/sec: 611.176
INFO:tensorflow:loss = 0.0703291, step = 401
INFO:tensorflow:global_step/sec: 611.198
INFO:tensorflow:loss = 0.0639818, step = 501
INFO:tensorflow:global_step/sec: 603.184
INFO:tensorflow:loss = 0.0596184, step = 601
INFO:tensorflow:global_step/sec: 608.977
INFO:tensorflow:loss = 0.0564216, step = 701
INFO:tensorflow:global_step/sec: 609.325
INFO:tensorflow:loss = 0.0539899, step = 801
INFO:tensorflow:global_step/sec: 609.581
INFO:tensorflow:loss = 0.0520836, step = 901
INFO:tensorflow:global_step/sec: 611.075
INFO:tensorflow:loss = 0.0504887, step = 1001
INFO:tensorflow:global_step/sec: 609.998
INFO:tensorflow:loss = 0.049227, step = 1101
```

```
INFO:tensorflow:global_step/sec: 614.02
INFO:tensorflow:loss = 0.0481185, step = 1201
INFO:tensorflow:global_step/sec: 611.547
INFO:tensorflow:loss = 0.0471658, step = 1301
INFO:tensorflow:global_step/sec: 614.462
INFO:tensorflow:loss = 0.0462878, step = 1401
INFO:tensorflow:global_step/sec: 612.448
INFO:tensorflow:loss = 0.0454978, step = 1501
INFO:tensorflow:global_step/sec: 607.526
INFO:tensorflow:loss = 0.0447773, step = 1601
INFO:tensorflow:global_step/sec: 616.512
INFO:tensorflow:loss = 0.0440958, step = 1701
INFO:tensorflow:global_step/sec: 616.203
INFO:tensorflow:loss = 0.0434451, step = 1801
INFO:tensorflow:global_step/sec: 622.39
INFO:tensorflow:loss = 0.0428362, step = 1901
INFO:tensorflow:Saving checkpoints for 2000 into
/tmp/tmpk_kAvv/model.ckpt.
INFO:tensorflow:Loss for final step: 0.0425141.
```

INFO 信息是每迭代 100 次输出一次，包括所用计算时间及目标损失函数的值。整体上，损失函数的值一直在下降，只是幅度在逐渐放缓；在前 300 轮，损失函数的值下降很快。

模型训练完毕，可以使用 DNN 分类器的计算评估值功能，指定验证数据集的特征和分类值，并打印输出其精确度。

```
    accuracy_score = classifier.evaluate( x=test_set.data,
y=test_set.target)["accuracy"]
    print("Accuracy: {0:f}".format(accuracy_score))
```

打印输出的结果如下，即 Accuracy（精确度）为 0.96667，与上节的 softmax 模型的精确度相等。

```
Accuracy: 0.966667
```

上节训练出 softmax 模型后，我们自定义了两条数据 new_samples 进行预测，这里同样可以使用深度学习模型预测 new_samples，并将预测结果打印输出。具体代码如下：

```
    y = list(classifier.predict(new_samples))
    print("Predictions: {}".format(str(y)))
```

打印输出的预测结果如下，预测第一条记录的类别索引号为 1，第二条的类别索引号为 2。与上节的 softmax 模型的预测结果相同。

```
Predictions: [1, 2]
```

附录 A

A.1 免费体验

用户可以在阿里云机器学习平台官网（https://data.aliyun.com/product/learn）申请公测，免费体验。官网页面如图 A-1 所示，“申请公测”按钮位于机器学习平台介绍的左下方。



图 A-1 阿里云机器学习平台官网

点击“申请公测”后，会出现如图 A-2 所示的登录界面，需要输入阿里云账号（如果没有

的话，您可以免费注册一个）。

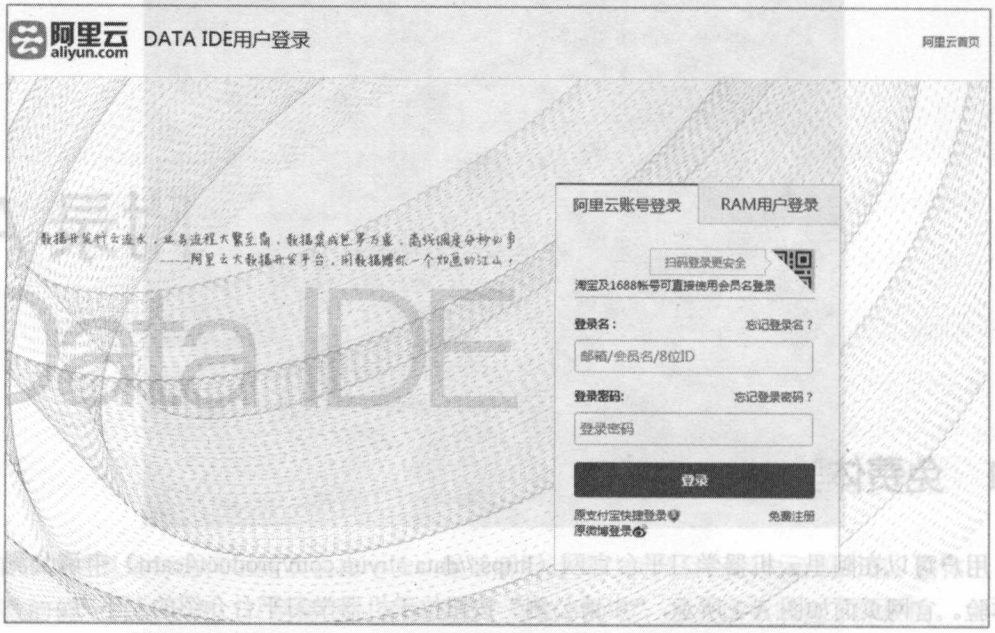


图 A-2 “免费体验”登录界面

登录成功后，系统会给您分配体验账号，包括子用户名称和子用户密码；然后，按照指引，用体验帐号登录“数加”平台。第一次使用，会提示你选择租户及工作空间，如图 A-3 所示，不需要输入内容，直接点击“提交”按钮即可。

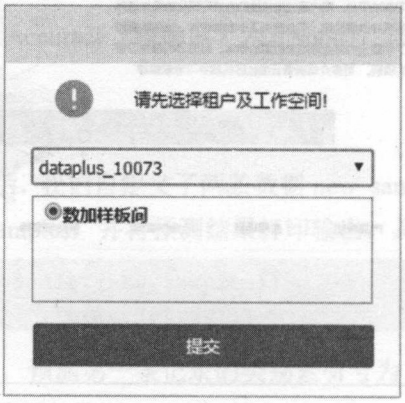


图 A-3 选择租户及工作空间

这样，我们就进入了“阿里云机器学习”的操作界面，如图 A-4 所示。最左边黑色背景的竖栏，包括了平台的 4 个重要元素：实验、数据源、组件和模型；点击任意图标，在其相邻的竖栏会显示与之相关的各种资源；右边的区域为工作区。

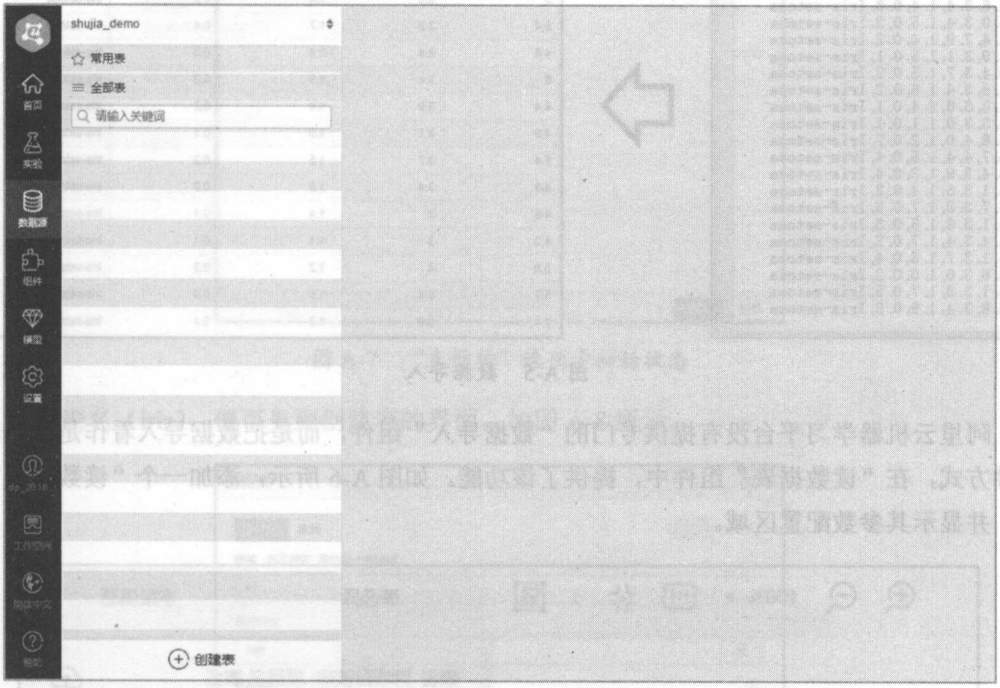


图 A-4　平台操作界面

A.2　数据导入

将 txt、csv 文件保存的结构化数据，导入到 MaxCompute，成为新的数据表，便于后面的分析。如图 A-5 中左边所示的 iris.txt 文本文件，内容是用换行符和逗号分隔的数据；对每一列数据命名，并指定类型，导入到 MaxCompute，数据表如图 A-5 所示。

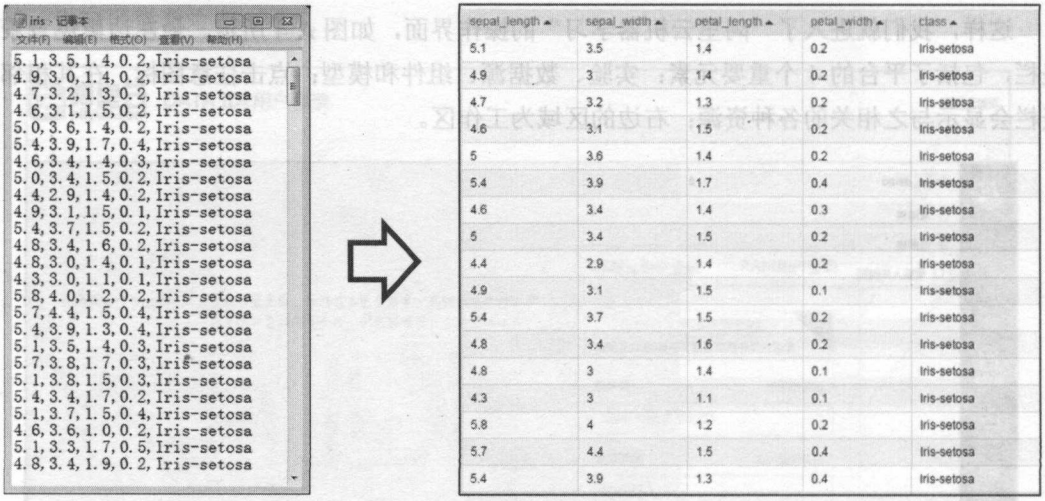


图 A-5　数据导入

阿里云机器学习平台没有提供专门的“数据导入”组件，而是把数据导入看作是创建表的一种方式，在“读数据表”组件中，提供了该功能。如图 A-6 所示，添加一个“读数据表”组件，并显示其参数配置区域。

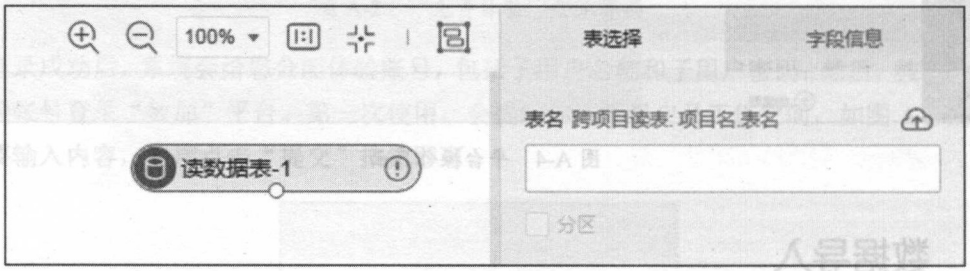


图 A-6　“读数据表”组件

点击图标，得到“创建表”的界面。首先看到的是“表结构”标签显示的内容，需要在此输入表名及表结构的信息，如图 A-7 所示。

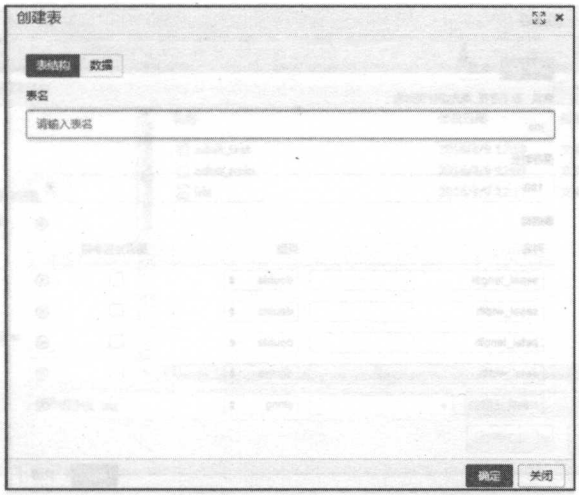


图 A-7 “表结构”选项卡初始状态

填写表名（iris），便可见到创建表的界面，如图 A-8 所示。



图 A-8 “数据”选项卡中添加表名后的状态

由于该表不存在，出现表结构设计区域，如图 A-9 所示，可以定义列名、列类型，并可以指定是否把该字段当“分区字段”看待。



图 A-9 表结构设计区域

随后，点击“数据”标签，进行数据导入。

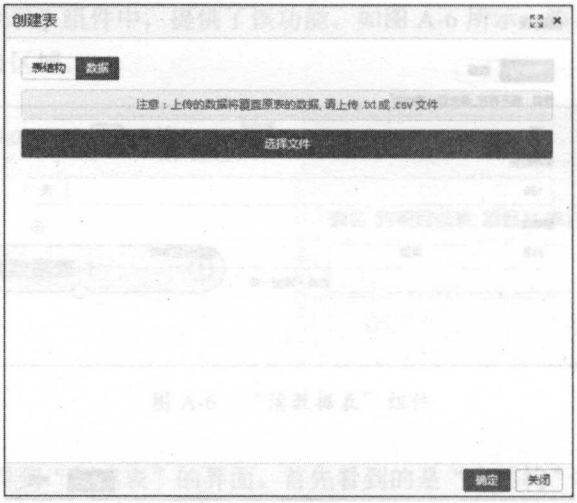


图 A-10 “数据”选项卡

点击“选择文件”按钮，出现文件选择窗体，如图 A-11 所示，选择文本文件 iris.txt。

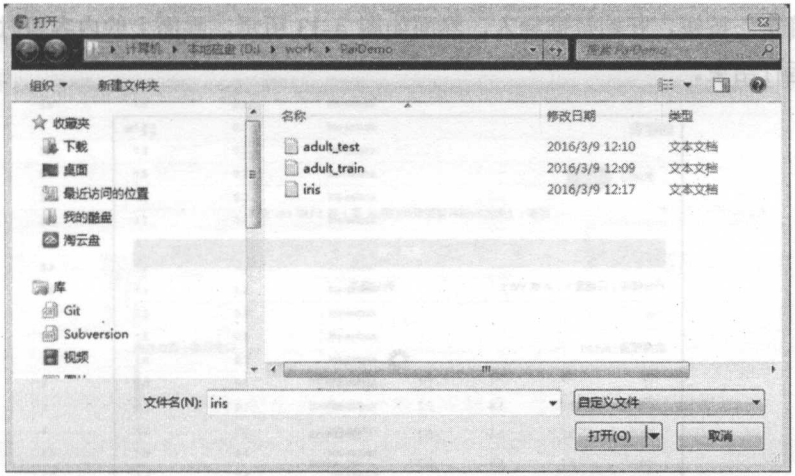


图 A-11　上传数据文件选择

该界面会自动给出建议的行分隔符、列分隔符，并显示分隔后的数据预览，如图 A-12 所示。



图 A-12　数据浏览

点击“确定”按钮，开始数据导入，界面如图 A-13 所示，界面上的内容暂时不允许修改、执行，直到运行结束。

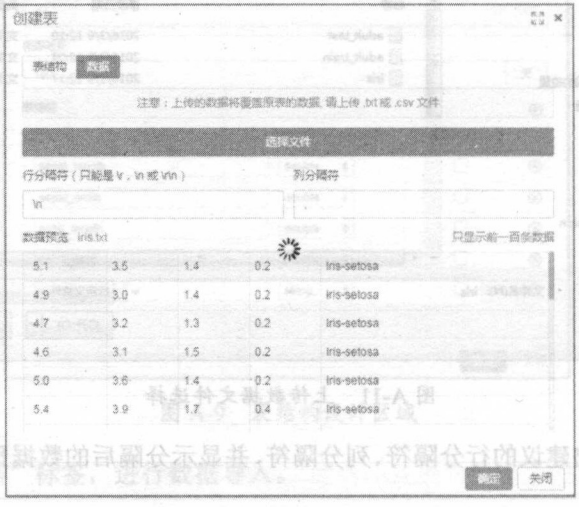


图 A-13　数据上传中

上传成功后，会回到主界面，如图 A-14 所示，显示该“读数据表”组件，并显示刚才导入的表名。

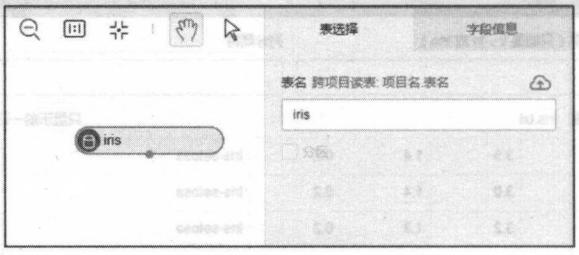


图 A-14　上传数据表成功

从该组件的弹出菜单中选择“查看数据”，显示如图 A-15 所示。

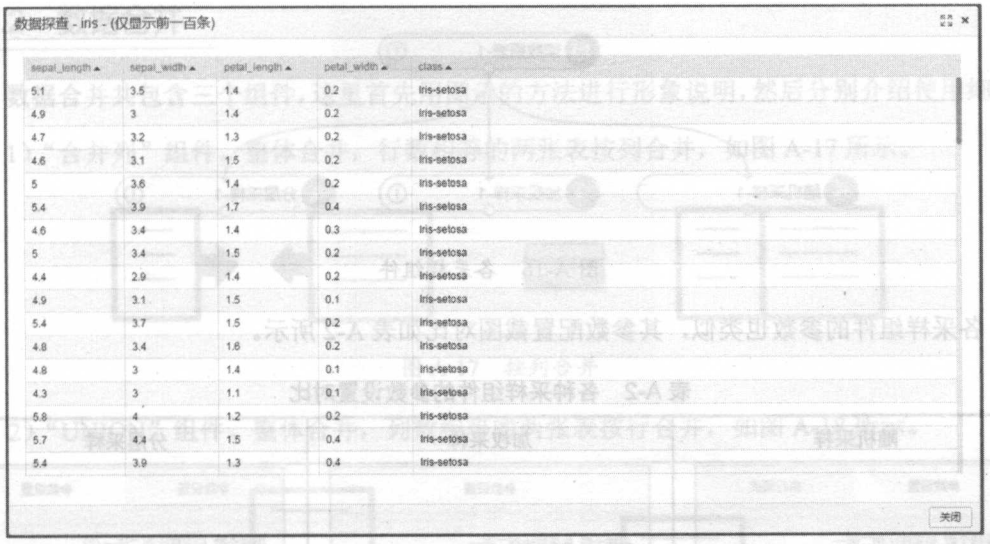


图 A-15　显示数据表内容

A.3　数据预处理

A.3.1　采样

有三种常用的采样方式：随机采样、加权采样和分层采样。具体描述详见表 A-1。

表 A-1　常用采样方法

名　称	描　述
随机采样	所有数据样本被选中的概率是一样的
加权采样	每个数据样本都有一个权值，该权值与被选中的概率成正比。譬如：对于样本 1 和样本 2，权值分别为 50 和 10，则样本 1 被选中的概率是样本 2 的 5 倍
分层采样	每个数据样本都有一个“标签”，可以据此将数据“分层”。可为每个标签（即其对应的一“层”数据）分别设置采样的比例。在处理分类问题时，对于正负样本数量差别悬殊的情况，常用该采样方法，对数量较多的“标签”设置较小的采样比例

每个采样组件都只有一个输入端，一个输出端，如图 A-16 所示。

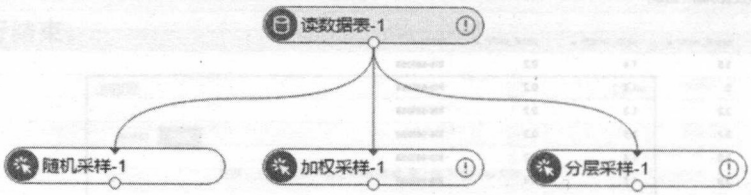


图 A-16　各采样组件

各采样组件的参数也类似，其参数配置截图对比如表 A-2 所示。

表 A-2　各种采样组件的参数设置对比

随机采样	加权采样	分层采样
参数设置　执行调优 采样个数 与采样比例二选一 采样比例 范围(0,1) 与采样个数二选一 放回采样 随机数种子 正整数 默认为空	参数设置 采样个数 与采样比例二选一 采样比例 范围(0,1) 与采样个数二选一 放回采样 权重列 支持double型和bigint型 随机数种子 正整数 默认为空	字段设置　参数设置 采样个数 与采样比例二选一 采样比例 范围(0,1) 与采样个数二选一 随机种子值 可选 1234567 并发计算数 可选 默认自动计算 每个核使用内存大小 可选 默认自动分配

由表 A-2 对比可知：

1）每种采样方式都可设置“采样个数”或“采样比例”，但不能同时设置。

2）每种采样方式都有参数“随机数种子”。设定相同的“随机数种子”，可以保证采样结果相同；不指定的时候，会将当前时间作为种子。

3）“放回采样”为可选项。放回采样是指逐个抽取样本时，选中的样本还会被放回总体中，随后可能被多次选中。不放回采样则可保证每个样本最多被选中一次。

A.3.2 数据合并

数据合并共包含三个组件，这里首先用图示的方法进行形象说明，然后分别介绍使用细节。

1）“合并列”组件。整体合并，行数相等的两张表按列合并，如图 A-17 所示。

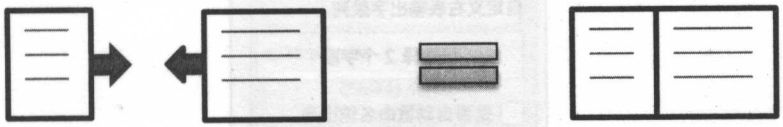


图 A-17　按列合并

2）“UNION”组件。整体合并，列数相等的两张表按行合并，如图 A-18 所示。

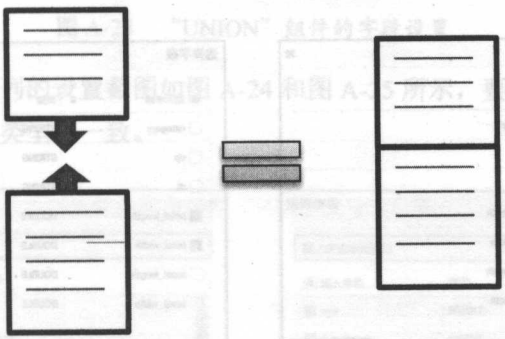


图 A-18　按行合并

3）“JOIN”组件。逐个记录合并，按某个键值将两个表中的记录关联起来，如图 A-19 所示。

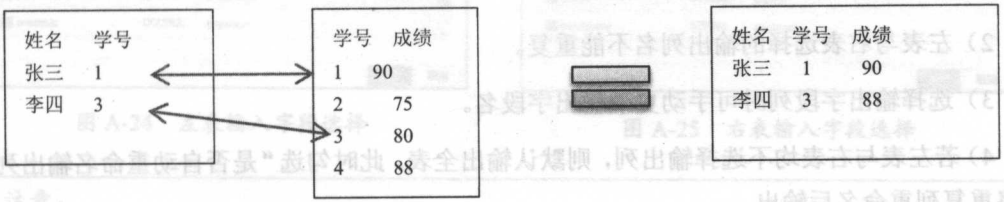


图 A-19　记录关联

1. 合并列

将两张表的数据按列合并，需要表的行数保持一致。参数设置如图 A-20 所示。

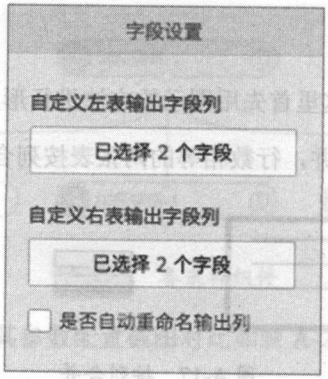


图 A-20 “合并列”组件的参数设置

其中，自定义的左表、右表输出字段列的截图，如图 A-21 和图 A-22 所示。

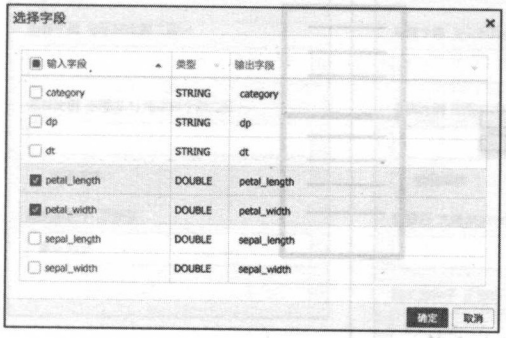


图 A-21 左表输入字段选择

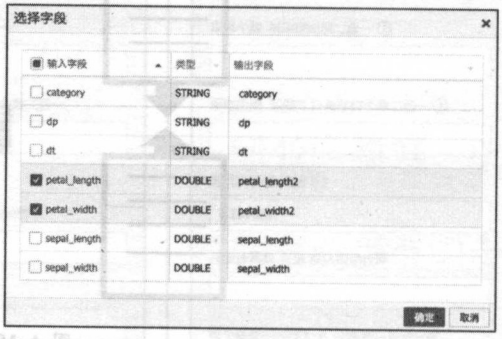


图 A-22 右表输入字段选择

使用时需要注意：

1）选择的两张表行数需保持一致。

2）左表与右表选择的输出列名不能重复。

3）选择输出字段列时可手动更改输出字段名。

4）若左表与右表均不选择输出列，则默认输出全表。此时勾选“是否自动重命名输出列”，会将重复列重命名后输出。

2. UNION

将两张表的数据按行合并，左表及右表选择输出的字段个数以及类型应保持一致。整合了 union 和 union all 的功能。参数设置如图 A-23 所示。

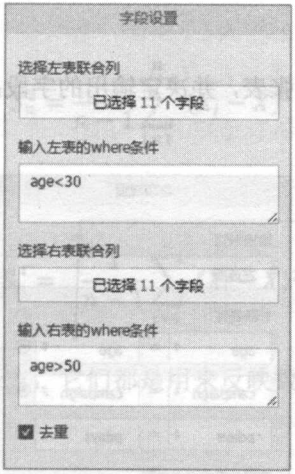


图 A-23 “UNION”组件的字段设置

其中，左、右表联合列的设置截图如图 A-24 和图 A-25 所示，要保证左右表选择的列数需相同，每个输出的列名及类型要一致。



图 A-24 左表输入字段选择



图 A-25 右表输入字段选择

注意：

- 可根据实际情况在条件框中手动输入已选字段的过滤条件，默认情况下是选择全表，目前操作符支持“=”，“!=”，“>”，“<”，“>=”，“<=”，“like”和“rlike”。
- 系统默认勾选“去重”。勾选后将会对生成的数据表的重复行进行去重操作（distinct）右表联合列。

3. JOIN

两张表通过关联信息，合成一张表，并决定输出的字段；与 SQL 的 join 语句功能类似。参数设置截图如图 A-26 所示。



图 A-26 “JOIN”组件的字段设置

注意：

- 连接类型支持：左连接，内连接，右连接，全连接。
- 关联条件目前只支持等式。
- 可手动添加或删除关联条件。

A.3.3 标准化、归一化、缺失值填充

我们先回顾几个统计概念。对于数据$X_1, X_2, \cdots, X_n$而言：

均值（Mean）

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

用来描述数据取值的平均位置。

方差（Sample Variance）

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

标准差（Standard Deviation）

$$S_n = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

标准差在数值上等于方差的开方，它们都是用来反映数据取值的离散（变异）程度。标准差的量纲与数据的量纲相同。

最大值（Maximum）

$$X_{max} = max(X_1, X_2, \cdots, X_n)$$

最小值（Minimum）

$$X_{min} = min(X_1, X_2, \cdots, X_n)$$

1．标准化

标准化操作的定义很简单，一个表的某一列数据就是$X_1, X_2, \cdots, X_n$，对其进行变换

$$X_i' = \frac{X_i - \bar{X}}{S_n}$$

得到的数据$X_1', X_2', \cdots, X_n'$就是标准化处理的结果，将此结果数据存入结果表中。

标准化操作的结果，也称作标准分数（Standard Score）或 Z 分数（Z Score）。X_i'代表着X_i和均值$\bar{X}$之间的距离，是以标准差S_n为单位的。

对一个服从一般正态分布的随机变量$X \sim N(\mu, \sigma^2)$，使用标准化操作转换成

$$Z = \frac{X - \mu}{\sigma}$$

则，Z为一个服从标准正态分布的随机变量，即$Z \sim N(0, 1)$。

标准正态分布的密度分布函数如图 A-27 所示，有一些特点值得注意：

1）密度函数关于平均值对称，平均值与它的众数（statistical mode）以及中位数（median）为同一数值，标准差为 1。

2）68.268949%的面积在平均数左右的一个标准差范围内，即区间$[-1,1]$。

3）95.449974%的面积在平均数左右的两个标准差范围内，即区间$[-2,2]$。

4）99.730020%的面积在平均数左右的三个标准差范围内，即区间$[-3,3]$。

5）99.993666%的面积在平均数左右的四个标准差范围内，即区间$[-4,4]$。

我们遇到的很多数据都近似服从正态分布，则在那些 Z Score 的绝对值很大的时候，是异常点的可能性也很大。

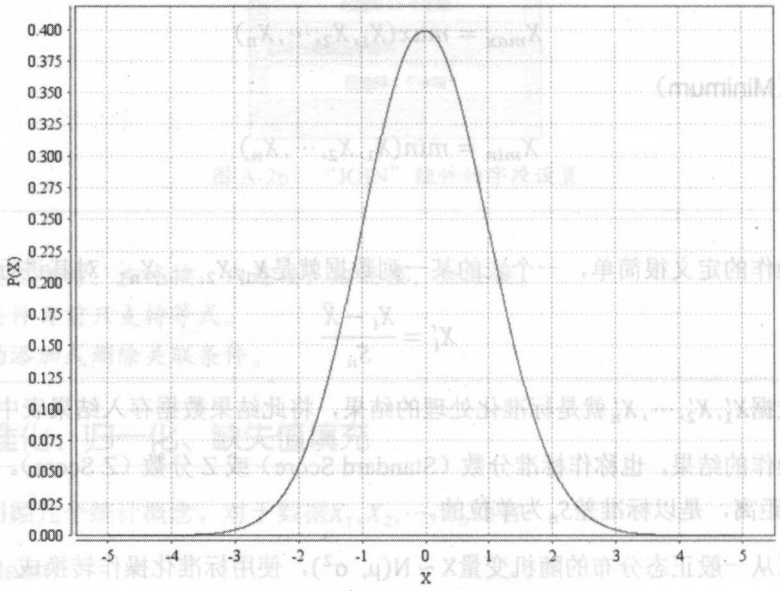


图 A-27　标准正态分布的密度分布函数

2．归一化

归一化的命名比较形象。一个表的某一列数据就是$X_1, X_2, \cdots, X_n$，对其进行变换

$$X_i' = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

由最大值、最小值的定义，可知

$$0 \leq X_i - X_{min} \leq X_{max} - X_{min}$$

所以，$0 \leq X_i' \leq 1$，即得到的数据$X_1', X_2', \cdots, X_n'$都在 0 与 1 之间，这就是“归一”化处理的结果，再将此结果数据存入结果表中。

3. 缺失值填充

缺失值填充用来将空值或者一个指定的值替换为最大值、最小值、均值或者一个自定义的值，可以使用如下的填充方法。

- 通过给定一个缺失值的配置列表，来实现将输入表的缺失值用指定的值来填充。
- 将数值型的空值替换为最大值，最小值，均值或者一个自定义的值。
- 将字符型的空值，空字符串，空值和空字符串，指定值替换为一个自定义的值。
- 待填充的缺失值可以选择空值或空字符，也可以自定义。
- 缺失值若选择空字符，则填充的目标列应是 string 型。
- 数值型替换可以自定义，也可以直接选择替换成数值最大值，最小值或者均值。

4. 使用技巧

这三个功能的组件比较相似，如图 A-28 所示。

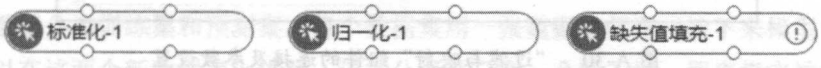


图 A-28　标准化、归一化、缺失值填充组件

这三个组件都含有两个输入桩和两个输出桩，定义方式也相似：

1）左边的输入桩对应数据输入表，即要进行数据预处理的表。

2）左边的输出桩为数据输出表，即经过数据预处理后产生的结果数据表。

3）右边的输入桩可输入配置表。

4）右边的输出桩可输出配置表。

这里的“配置表”，会根据不同的操作，记录不同的参数。标准化操作会记录各数据列的均值和标准差信息；归一化操作会记录各数据列的最大值和最小值信息；缺失值填充操作会记录各数据列需要填充的缺失值信息。通过此“配置表”，可以将某一数据表的操作应用到其他的数

据表上。一个典型的使用方式如图 A-29 所示。

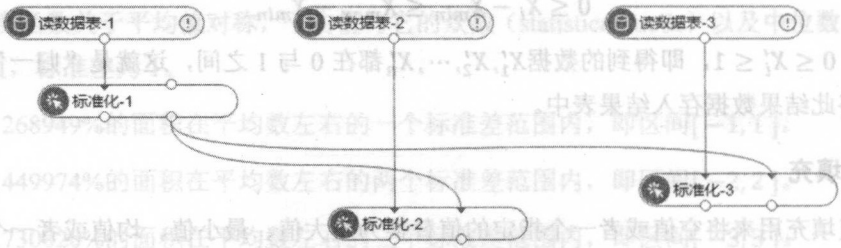


图 A-29　标准化组件配置表的生成及使用

13.3.4　过滤与映射

数据的过滤和映射可以在同一个组件内完成。选择该组件，并在输入桩连入一个数据源，如图 A-30 所示。

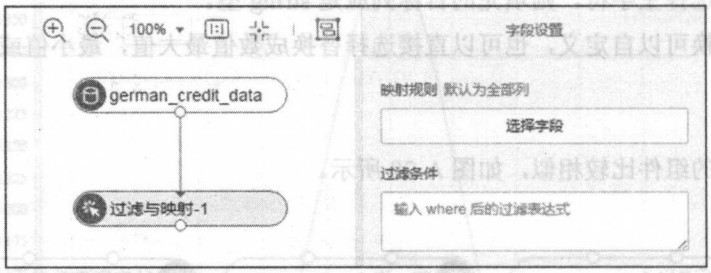


图 A-30　“过滤与映射”组件的连接及参数设置

图中相关概念解释如下。

- 过滤条件，就是 SQL where 语句后面的条件表达式。
- 映射规则，可以看作是在数据列上进行选择，只有被选中的列的数据才能保存到结果表中，默认是选择全部列。另外，还可以重命名保留到结果表中的数据列。操作界面如图 A-31 所示。



图 A-31 “过滤与映射”组件中映射规则的设置

13.3.5 数据拆分

数据拆分，是指将一个数据表按用户指定的比例，拆分成两个新的数据表。可以使用此组件，将原始数据分为训练集和预测集，每个数据集用一张数据表保存，接下来模型训练、预测和评估就可以在这两个新数据表上进行。该拆分组件只有一个输入桩，用来指定被拆分的原始数据集；有两个输出桩，分别输出拆分后的数据表；参数只有一个，就是结果表 1 所占的数据比例。如图 A-32 所示。

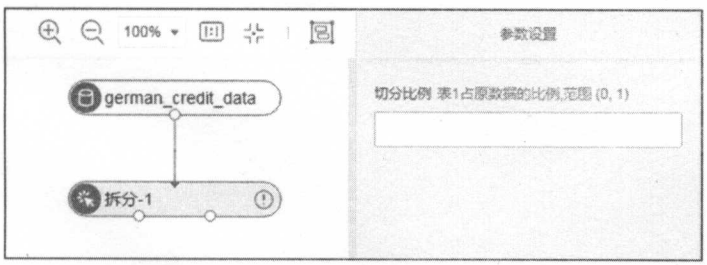


图 A-32 “拆分”组件的连接和设置

Broadview®
www.broadview.com.cn
博文视点·IT出版旗舰品牌
技术凝聚实力·专业创新出版

机器学习在线
解析阿里云机器学习平台



阿里云机器学习平台上集合了近百种算法组件，使用者不必掌握艰深的理论知识，即可通过对组件的使用而完成相关的任务。

本书的重点放在算法的使用上，通过实际的数据和具体的场景，帮助读者理解各类算法所擅长处理的问题；同时在内容上是根据机器学习的知识点由浅入深来逐层讲述的，这样降低了阅读的门槛，让读者对所学的内容能产生清晰的印象，并熟练地运用到实践中去。



注册博文视点社区（www.broadview.com.cn）用户，即享受以下服务：

提勘误赚积分：可在【提交勘误】处提交对内容的修改意见，若被采纳将获赠博文视点社区积分（可用来抵扣购买电子书的相应金额）。

交流学习：在页面下方【读者评论】处留下您的疑问或观点，与作者和其他读者共同交流。

页面入口：http://www.broadview.com.cn/31869

博文视点Broadview

新浪微博 weibo.com
@博文视点Broadview

策划编辑：刘　皎　@皎丫子
责任编辑：郑柳洁
封面设计：侯士卿

欢迎投稿
Ljiao@phei.com.cn
010-88254395

上架建议：大数据

ISBN 978-7-121-31869-6

定价：79.00元